Facial Micro- and Macro-Expression Spotting and Generation Methods

C H YAP

PhD 2022

Facial Micro- and Macro-Expression Spotting and Generation Methods

CHUIN HONG YAP

A thesis submitted in partial fulfillment of the requirements of Manchester Metropolitan University for the degree of Doctor of Philosophy

Department of Computing and Mathematics
Faculty of Science and Engineering
Manchester Metropolitan University

2022

Abstract

Facial micro-expression (ME) recognition requires face movement interval as input, but computer methods in spotting ME are still underperformed. This is due to lacking large-scale long video dataset and ME generation methods are in their infancy. This thesis presents methods to address data deficiency issues and introduces a new method for spotting macro- and micro-expressions simultaneously.

This thesis introduces SAMM Long Videos (SAMM-LV), which contains 147 annotated long videos, and develops a baseline method to facilitate ME Grand Challenge 2020. Further, a reference-guided style transfer of StarGANv2 is experimented on SAMM-LV to generate a synthetic dataset, namely SAMM-SYNTH. The quality of SAMM-SYNTH is evaluated by using facial action units detected by OpenFace. Quantitative measurement shows high correlations on two Action Units (AU12 and AU6) of the original and synthetic data.

In facial expression spotting, a two-stream 3D-Convolutional Neural Network with temporal oriented frame skips that can spot micro- and macro-expression simultaneously is proposed. This method achieves state-of-the-art performance in SAMM-LV and is competitive in CAS(ME)^2, it was used as the baseline result of ME Grand Challenge 2021. The F1-score improves to 0.1036 when trained with composite data consisting of SAMM-LV and SAMM-SYNTH. On the unseen ME Grand Challenge 2022 evaluation dataset, it achieves F1-score of 0.1531.

Finally, a new sequence generation method to explore the capability of deep learning network is proposed. It generates spontaneous facial expressions by using only two input sequences without any labels. SSIM and NIQE were used for image quality analysis and the generated data achieved 0.87 and 23.14. By visualising the movements using optical flow value and absolute frame differences, this method demonstrates its potential in generating subtle ME. For realism evaluation, the generated videos were rated by using two facial expression recognition networks.

Declaration

This thesis is submitted to Manchester Metropolitan University in support of my application for admission to the degree of Doctor of Philosophy. No part of this thesis has been submitted in support of an application for another degree or qualification of this or any other institution of learning. Work relating to this thesis has appeared in the following publications:

1. Chuin Hong Yap, Connah Kendrick, and Moi Hoon Yap, “Samm long videos: A spontaneous facial micro-and macro-expressions dataset,” in 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), IEEE, 2020, pp. 771–776.

2. Chuin Hong Yap, Moi Hoon Yap, Adrian K. Davison, Connah Kendrick, Jingting Li, Sujing Wang, and Ryan Cunningham, “3d-cnn for facial micro- and macro-expression spotting on long video sequences using temporal oriented reference frame,” in Proceedings of the 30th ACM International Conference on Multimedia, pp. 7016-7020. 2022.

3. Chuin Hong Yap, Ryan Cunningham, Adrian K. Davison, and Moi Hoon Yap, “Synthesising facial macro- and micro-expressions using reference guided style transfer,” Journal of Imaging, vol. 7, no. 8, p. 142, 2021.

Acknowledgements

Throughout my PhD and preparation of this thesis, many people have guided my understanding and knowledge that allowed me to form the work presented. First, I would like to show my utmost gratitude to my principal supervisor, Prof. Moi Hoon Yap, for your kindness, patience and support throughout my PhD project. You are an inspiring and supportive person to my well-being, especially when dealing with failures and rejections. I would like to sincerely thank my first supervisor, Dr. Ryan Cunningham, for your guidance, patience and knowledge of deep learning. I appreciate every moment we spent discussing potential ways to improve our algorithm and take our research to the next level. I would like to thank my second supervisor, Dr. Adrian K. Davison, for your experience in affective computing and facial action coding system. I learnt a lot during our discussion and it widens my knowledge in psychology. I would also like to thank my third supervisor, Prof. Darren Dancey, for providing career guidance and professional development opportunities.

I would like to thank my family for being supportive all the time throughout my life regardless of good or bad times.

I have received help and advice, directly or indirectly, from my peers and colleagues throughout my study. These people motivated me to share ideas and build relationships to further expand my knowledge. I want to thank Dr. Connah Kendrick for his knowledge and experience in programming and deep learning. I will miss the lame jokes and dabs we made during our time in the office. I would also thank every member of E121: Mr. Bill Cassidy, Dr. Sean Barton, Dr. Jhan Alarifi, Dr. Jireh Jam, Dr. Xulu Yao, Dr. Guido Ascenso and Mr. Hongyuan Xie. You all are the best peers that I could have ever asked for.

I would like to thank Prof. Yonghong Peng and Centre for Advanced Computational Science

(CfACS) for the financial support that I received for attending seminars and conferences during my PhD.

I am grateful to Manchester Metropolitan University for providing the studentship for me to complete this thesis and provide valuable experience in teaching at a university level.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Facial expression is movements of one or more muscles of the face. It is a type of nonverbal communication which may indicate the emotional state of a person. However, facial expression is not a perfect representation of the emotional state of a person as humans can hide their emotions by deliberately posing certain facial expressions (e.g. faking a smile). In this thesis, regular facial expressions are called facial macro-expression (MaE); while there exist subtle facial movements which may happen during emotion suppression which is called facial micro-expression (ME). Our research focus is on facial micro-expression, a subtle facial movement that is involuntary and occurs less than 0.5s [1]. These expressions occur less frequent compared to MaE counterparts and happen more in stressful or high stake situations.

Facial analysis is a rapidly growing field, especially with the advent of deep learning which accelerates this research area. These include face detection, face recognition, facial expression recognition, age estimation and gaze estimation. It converts raw footage or image of facial expressions into meaningful features or data. Subsequently, these data can be evaluated by humans or machines to obtain a deeper understanding of the facial structure or movements and enable a wide range of security applications (i.e. face unlock), psychological counselling, interrogation and medical purposes. There exists software and tools [2]–[4] used in face-related research that primarily tracks the face based on detected features (e.g. facial landmark and facial action units). However, there is no existing software for ME spotting.

Deep learning is the state-of-the-art approach in various fields (i.e. computer vision and natural language processing) including face-related research. Machine learning model performs the best in ME recognition [5] uses a multi-stream neural network. This multi-stream neural network takes in 3 different extracted features (3 optical flow based features) as input. It demonstrated the use of 3 different extracted features increases the number of input features which enables the network to capture more information. The main advantage of deep learning model is the performance of the model improves as the number of data increases. By introducing more data with higher diversity, the model is capable of generalising on unseen data better. However, we identified the main limitations of this research which are the limited number of available ME dataset and the imbalance of emotion classes of labelled data. In the current stage of development, ME recognition and spotting research are only conducted in laboratory-controlled environments online. There exist a dataset curated from spontaneous videos [6] of high stake situations, but this dataset contains only 31 videos of 16 individual and is only used as an evaluation tool rather than training. As a result, the real-world in-the-wild application of this technology in the current stage is still infeasible due to the reason mentioned.

To make ME analysis into a real-life application, the analysis algorithm should be able to differentiate and detect macro- and micro-expression simultaneously. There is documented evidence that shows overlaps between facial macro- and micro-expressions [7] which strongly highlights the importance of this ability in coping with real-world data. Prior to this thesis, as far as we are aware, all algorithms do not have the ability to spot both simultaneously which presents a research gap that we are interested in. The ability to spot both expressions at the same time can provide a more accurate overview of the dynamic of the overlap of macro- and micro-expression.

The goal of this thesis is to present our research attempt in addressing the data deficiency problem and creating a fully automated machine learning approach that can perform macro- and micro-expression spotting simultaneously. In order to achieve this goal, we curate a new long video dataset and introduce a synthetic dataset (Chapter 4), propose a machine learning approach that leverage the duration difference between macro- and micro-expressions (Chapter 5), design generative models that can potentially increase the number of available data pool (Chapter 6) and two facial expression recognition networks to quantify the quality of the facial expressions generated in Chapter 6 (Chapter 7).

1.2 Motivation

Deep learning is the state-of-the-art in many research areas. This method requires a large-scale and diverse data to achieve its full potential. Existing methods are not able to spot both macro- and micro-expression simultaneously. In the current stage of micro-expression research, there is a lack of dataset and many researchers are focused on ME recognition, which requires manual human labelled short video clips as input. Additionally, there is no available method to spot ME and MaE simultaneously. The motivation of this thesis are:

1. To improve deep learning based ME spotting by introducing more datasets with higher variety to improve the generality of the model.

2. To enhance the ME spotting performance by introducing end-to-end like method that can simultaneously spots both ME and MaE. This can eliminate the reliance on external algorithms or models that contains certain biases.

3. To bridge the gaps in ME research towards automated ME and MaE spotting and serve as baseline methods for international ME Challenges to facilitate future research.

1.3 Problem Statement

We identified a few key research gaps in ME research. First, the lack of long videos dataset. Most of the existing datasets are short video clips and do not represent real-world scenario as MEs are not common and can occur simultaneously alongside with MaE. Next, there is no end-to-end approach and method that can spot ME and MaE simultaneously. Most methods has one or more pre- and post-processing stages, which potentially introduce dependency and error. These methods are not ideal in real-time setting. This thesis aims to bridge the gap by researching into new end-to-end deep learning techniques with minimal reliant on external procedures.

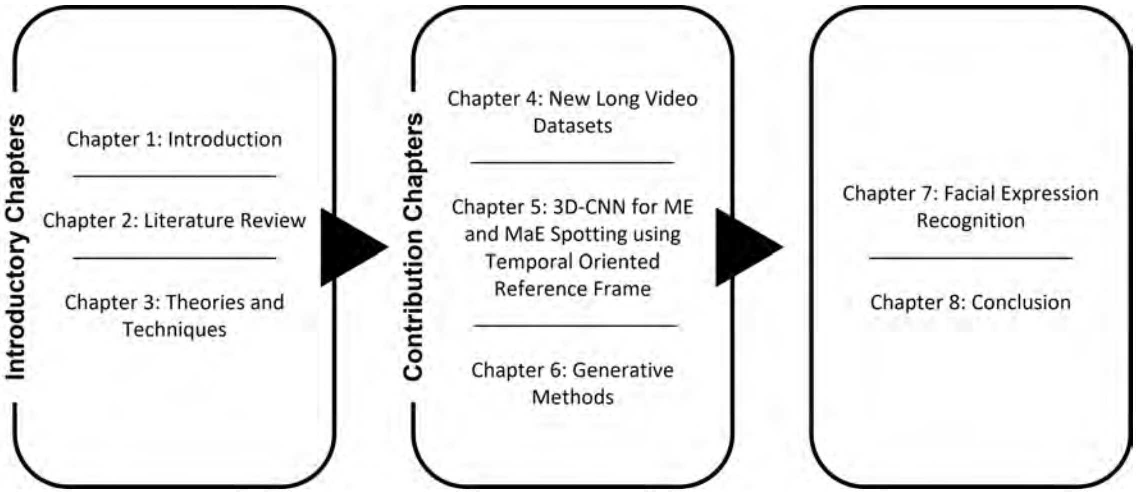

Introductory Chapters
Chapter 1: Introduction
Chapter 2: Literature Review
Chapter 3: Theories and Techniques
Contribution Chapters
Chapter 4: New Long Video Datasets
Chapter 5: 3D-CNN for ME and MaE Spotting using Temporal Oriented Reference Frame
Chapter 6: Generative Methods
Chapter 7: Facial Expression Recognition
Chapter 8: Conclusion

Figure 1.1: An overview visualisation of the structure of this thesis. The first three chapters includes the information about micro-expression research and the theories/techniques involved. The subsequent four chapters are the contributions of this thesis. The final chapter concludes this thesis with the summary and future work of this work.

1.4 Objectives

This project primarily aims to develop a spatial- and temporal-dimension based deep learning algorithm for micro-expression (ME) spotting and generation method. The objectives are:

1. To identify research gap and determine research direction through literature review.
2. To address data deficiency by introducing a new ME long video dataset.
3. To investigate existing GAN models in ME generation and define evaluation method to compare original data and generated data.
4. To propose a new ME spotting algorithm based on deep learning method and validate it on ME long videos datasets.
5. To design a new generative model that can generate spontaneous ME to improve the diversity of datasets in ME research.

1.5 Contributions

The main contribution of this thesis are:

1. A new long videos dataset containing both facial micro- and macro-expression with the highest face resolution. A benchmark result for ME and MaE spotting is introduced based on Facial Action Unit (AU) obtained using a facial analysis toolkit, OpenFace.
2. An advanced image augmentation method using style transfer to create more data variation on current existing dataset.
3. A deep learning facial micro- and macro-expression spotting algorithm using the temporal difference of facial micro- and macro-expression which spots both expressions simultaneously.
4. A spontaneous recursive generation model that is capable of generating a sequence of spontaneous facial expressions. This model can be applied to generate spontaneous micro-expressions.

1.6 Thesis Organisation

This thesis is organised by Chapter 2 as Literature Review, Chapter 3 as Theories and Techniques, the main contributions are in Chapter 4 to 7 and everything is summarised in Chapter 8 as Conclusion.

Chapter 2 introduces the basic knowledge and literature review on annotation system, micro-expression analysis methods and generative method centred on facial expression.

Chapter 3 is mainly about the technical details of steps, processes and methods that uses throughout the research. This includes preprocessing methods, feature descriptors, machine learning approaches and performance metrics.

Chapter 4 is attempts to increase the data pool of this research by creating more dataset. The first attempt is curating a new long video dataset, SAMM Long Videos, that contains both ME and MaE. Benchmark results for SAMM Long Videos is also introduced by taking advantage of the automated facial action units detected by a facial analysis toolkit. The second attempt is generating a synthetic dataset (SAMM-SYNTH) using reference-guided image synthesis based on SAMM Long Videos dataset. Both quantitative and qualitative measurement on the generated data was proposed.

Chapter 5 is a ME and MaE spotting method that is capable of spotting both expression simultaneously. This method leverage the duration difference between ME and MaE by using a two stream convolutional neural network with different frame skips.

Chapter 6 is generative models that can generate new spontaneous data. By using generative model, the number of data can be increased and the class imbalance in current dataset which is a known issue can potentially be resolved. Our method is a recursive generation method that is capable of generating spontaneous facial expression sequences without using any labels. This method is also capable of generating micro-movements and has potential in generalised onto other domain. An ambitious attempt at removing the constant reference frame is also discussed.

Chapter 7 contains facial expression recognition approaches that can be used as an evaluation method for the generated facial expressions. We demonstrate two different approach (using classification and regression) on this task and discussed the implications on facial expression sequences.

Chapter 8 concludes this thesis with the summary of the thesis and potential future works.

Chapter 2

Literature Review

This section reviews background knowledge on emotions, facial expressions, facial expression annotation, datasets, and micro-expression (ME) related works. The latter includes ME recognition, ME and MaE spotting, and generative models.

ME recognition is the classification of MEs into one of the emotion classes based on facial movements. There are evaluations which split emotions into negative, positive, and surprise, which is also the benchmark for the 2^{nd} Facial Micro-Expression Grand Challenge (MEGC) [8]. However, common approaches use the instance where a facial expression peaks in intensity, and the true expression is not known or labelled. As a result, ME spotting was introduced. ME spotting is the detection of ME in a time series sequence and is proposed to be the first step of a ME analysis pipeline, as it scans video sequences for ME presence before any further analysis. This could potentially reduce the number of false positives, a known issue in this research area. By locating MEs accurately in a sequence, further analysis errors, such as the algorithm classifying a stationary face as a ME, can be eliminated.

There is a limited amount of data in ME research. Generative models can be an alternative in obtaining more data by generating synthetic data that are similar to real-world data. This chapter also discusses facial expression generation and ME generation techniques that can potentially be a solution to this problem.

2.1 Emotions and Facial Expressions

Emotions are conscious, mental reactions towards a certain circumstance, thing or behaviour. It is a very complex phenomenon across different fields such as neuroscience, psychology, and social science. There exist several theories of emotion such as basic emotion theory [9], [10] (emotions are divided into discrete categories), emotion as a body-brain combination [11], and appraisal model [12], [13] (psychology state is based on human self-interpretation).

Facial expressions are one of the main visual cues of the human emotional state. They are quick and efficient as they communicate socially relevant information via the face, which is visible in most everyday scenarios. They are highly adaptive because they quickly mobilise and coordinate resources needed to successfully deal with life tasks, which is an advantage for survival.

2.1.1 Universal Emotions

Our research follows 7 universal basic emotions [14]: happy, sad, anger, disgust, fear, surprise, and contempt. The seventh emotion, contempt, is the most recent addition and has contradicting arguments against its existence [15]. The first person who proposed universal emotions is Charles Darwin [16]. This theory states there are a certain set of facial expressions that are universal across cultures, species, and are not learnt. Inspired by this, Tomkins [17] formulated Affect Theory, which focused on the importance of the face in conceiving different emotions. These theories influenced Ekman and Friesen [18] in their experiments in Papua New Guinea, where they show flashcards (containing the facial behaviour of one of six emotions) to the locals and record their responses. Although the local tribes had little to no contact with outsiders (no knowledge of any Westernised culture), they were able to recognise facial expressions and emotions proposed by Darwin and Tomkins. Ekman [19] later proposed there might be more than six/seven emotions which are more complex. We focus on basic emotion theory rather than this expansion.

The visualisation of posed expressions of the 7 basic emotions can be seen in Figure 2.1.

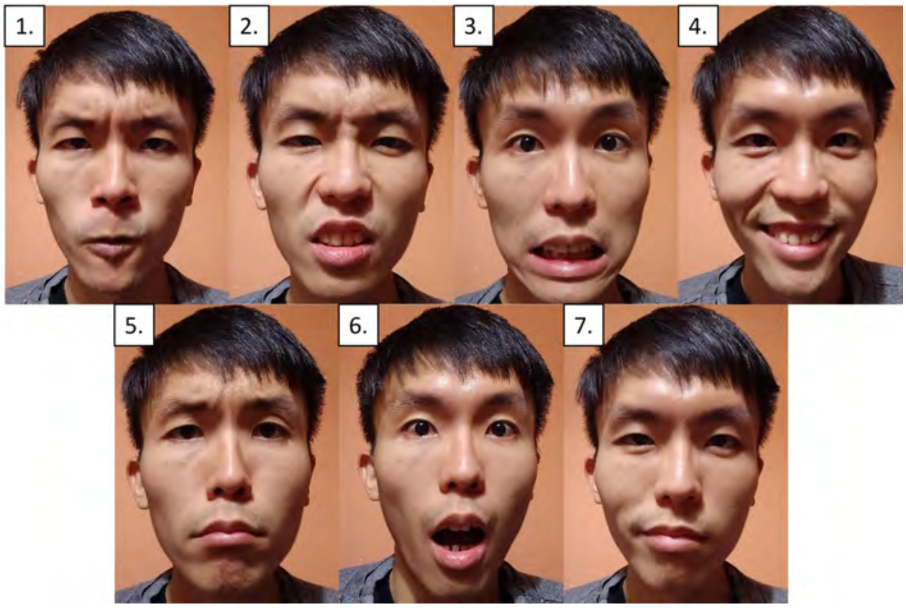


Figure 2.1: Posed facial expression of 7 universal basic emotions. Where 1 = Anger, 2 = Disgust, 3 = Fear, 4 = Happiness, 5 = Sadness, 6 = Surprise, 7 = Contempt.

2.1.2 Facial Micro- and Macro-expression

Facial expressions are facial muscle movements that are often associated with certain emotions. It is a type of non-verbal communication that usually conveys the emotional state of a person. There also exists some outlier cases where expression change is due to damage to the nervous system [20].

In micro-expression research, a clear distinction can be made between two types of facial expressions: macro-expression (MaE) and micro-expression (ME). MaE, or a regular facial expression, typically lasts between 0.5 to 4.0s [21]. It usually has high intensity movements of the face, which can be detected easier. On the contrary, MEs are involuntary, low intensity movements of facial muscles which occur in less than 0.5s [21]. They occur more frequently in stressful and high stake situations [22], [23]. Due to their involuntary nature, analysing MEs can reveal the emotional state of a person when they are attempting to conceal it [24].

However, MEs are more challenging to detect or classify, as these expressions are often too short and subtle for human eyes to spot. In addition, humans have innate psychological biases towards emotion [25], which is a disadvantage when it comes to expression classification. Machines are not susceptible to the same psychological biases as a human. This can be a useful trait for facial or ME expression analysis where the machine can provide an objective assessment when judging what facial expression is shown.

2.2 Facial Expression Annotation

Static image annotation and dynamic video annotation are fundamentally different in nature. Although they both shared the same Facial Action Unit Coding System annotation (which will be explained in the next subsection), video has an additional complexity of temporal information to be taken into account. Static images are commonly annotated using the peak of a facial expression. Video annotation includes different phases within a facial expression, consisting of onset (where the facial expression starts), apex (where the facial expression has the highest intensity), and offset (where facial expression ends).

2.2.1 Facial Action Coding System

The Facial Action Coding System (FACS) is one of the most established facial movement analysis for human coders to manually annotate facial movements based on the muscle movements that are categorised into each facial action unit (AU). Each AU is either constriction or relaxation of one or more muscles. There are also action descriptors, which may involve several muscle movements such as head movements (e.g. head turn right, head tilt left, etc.), eye movements (e.g. eyes turn left, eyes up, etc.), visibility of facial features (e.g. eyes not visible, lower face not visible, etc.), and gross behaviour (e.g. jaw movements, chewing, etc.). A few common AUs used are shown in Table 2.1, key AUs are visualised in Figure 2.3, and the related muscles are shown in Figure 2.4. For each basic emotion, the relevant AUs commonly associated are shown in Table 2.2.

2.2.2 Video or Image Sequence Annotation

For facial expression sequence annotation, the AU and the duration of the expression are taken into account. AU annotation includes the type and intensity of the AU involved similar to still image annotation. The temporal dynamic of facial expressions follows a common activation pattern described in Figure 2.2.

The *Neutral* phase is when there are no facial muscle movements, the *Onset* phase is when facial expression starts (muscle contraction begins and intensity increases), the *Apex* is when

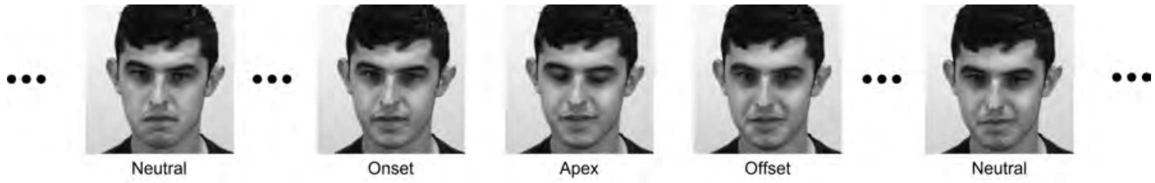


Figure 2.2: Example of facial expression sequence annotation. The temporal dynamic of facial expression is annotated with onset, apex and offset frames which represent the beginning, the peak and the end of the facial expression. The AUs involved are 12, 17, and 25. The images were taken from SAMM-LV dataset.

facial expression peaks (muscle contraction remains stable or plateaus and no longer increases), and the *Offset* is when the facial expression ends (muscle relaxation). Generally, all AUs follow this trend: facial expression onset from neutral phase, reaches a certain peak and offsets back to the neutral phase. However, in some videos, the full expression sequence is not captured (e.g. expression peaks at the beginning of the video, the expression offsets only halfway before the end of the video, visual occlusions etc.). There are also expressions that do not follow this pattern or overlap with other AUs.

2.3 Facial Micro-expression Datasets

In this section, datasets are categorised by short and long videos. Short video datasets contain mostly frames associated with MEs only, whereas long video datasets contain both MEs and MaEs. For ME spotting, detecting MEs on a short clip is much easier, but also unrealistic, as MEs do not occur that frequently. Long videos make the task more challenging as there are more non-ME movements and the spotting algorithm will need to try harder to find ME instead of movements. For earlier ME research, there were only short video datasets, while long video datasets were only produced in recent years. The main motivation for the creation of ME datasets is mainly due to the lack of accessible and well-established datasets for both ME recognition and spotting.

2.3.1 Short Videos Datasets

In early ME research, most datasets consisted of short video clips. The way that these datasets were created ensures the algorithm captures the important details of the sequence (MEs), as most clips start with an onset frame and end with an offset frame. These datasets are useful

Table 2.1: Common Action Units (AUs) [26] used for facial expression analysis.

AU	FACS name	Muscular basis
1	Inner brow raiser	frontalis (pars medialis)
2	Outer brow raiser	frontalis (pars lateralis)
4	Brow lowerer	depressor glabellae, depressor supercilii, corrugator supercilii
5	Upper lid raiser	levator palpebrae superioris, superior tarsal muscle
6	Cheek raiser	orbicularis oculi (pars orbitalis)
7	Lid tightener	orbicularis oculi (pars palpebralis)
8	Lips toward each other	orbicularis oris
9	Nose wrinkler	levator labii superioris alaeque nasi
10	Upper lip raiser	levator labii superioris, caput infraorbitalis
11	Nasolabial deepener	zygomaticus minor
12	Lip corner puller	zygomaticus major
13	Sharp lip puller	levator anguli oris (also known as caninus)
14	Dimpler	buccinator
15	Lip corner depressor	depressor anguli oris (also known as triangularis)
16	Lower lip depressor	depressor labii inferioris
17	Chin raiser	mentalis
18	Lip pucker	incisivii labii superioris and incisivii labii inferioris
19	Tongue show	-
20	Lip stretcher	risorius with platysma
21	Neck tightener	platysma
22	Lip funneler	orbicularis oris
23	Lip tightener	orbicularis oris
24	Lip pressor	orbicularis oris
25	Lips part	depressor labii inferioris, or relaxation of mentalis or orbicularis oris
26	Jaw drop	masseter; relaxed temporalis and internal pterygoid
27	Mouth stretch	pterygoids, digastric
28	Lip suck	orbicularis oris

Table 2.2: AUs commonly associated with each respective basic emotion. Note: ⋆ is not always considered a basic emotion.

Emotion	Action units
Happiness	6+12
Sadness	1+4+15
Surprise	1+2+5B+26
Fear	1+2+4+5+7+20+26
Anger	4+5+7+23
Disgust	9+15+17
Contempt⋆	R12A+R14A

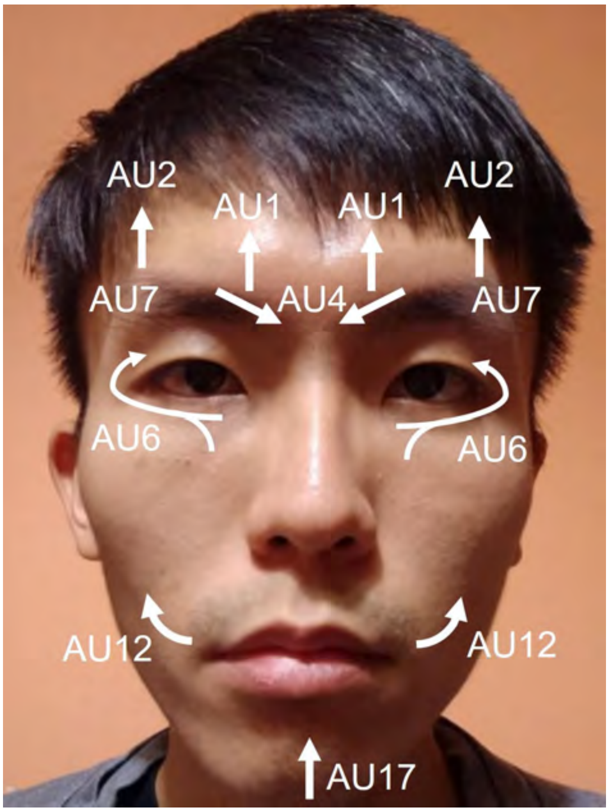


Figure 2.3: Key Action Units (AUs) visualised. This image was recreated from Li et al. [27].

for ME classification tasks.

Polikovsky: Polikovsky et al. [32] capture posed MEs using cameras with a resolution of 480×640 and a frame rate of 200 fps. 10 participants (5 Asian, 4 Caucasian, and 1 Indian) were asked to perform 7 basic emotions (disgust, anger, fear, sadness, happiness, surprise, and contempt) with low facial muscle intensity, and then revert to a neutral expression as soon as possible. The limitations of this dataset are that it contains posed MEs rather than spontaneous, which means it does not represent natural human behaviour. Next, the sample size is small and the ages of participants are similar with little variation as they are all university students. This dataset is not publicly available.

YorkDDT: YorkDDT [33] dataset consists of 20 videos, at 25 fps, and 320×240 resolution. This dataset is a study of deception detection. Participants were divided into two groups: the encoders and decoders. The encoders perform either truthful or deceptive actions and recorded them on video, while the decoders were asked to judge or indicate the truthfulness of each recording. The encoders consisted of 20 participants (8 males and 12 females). The decoders consisted of 30 participants (11 males and 19 females). The encoders were told to truthfully or

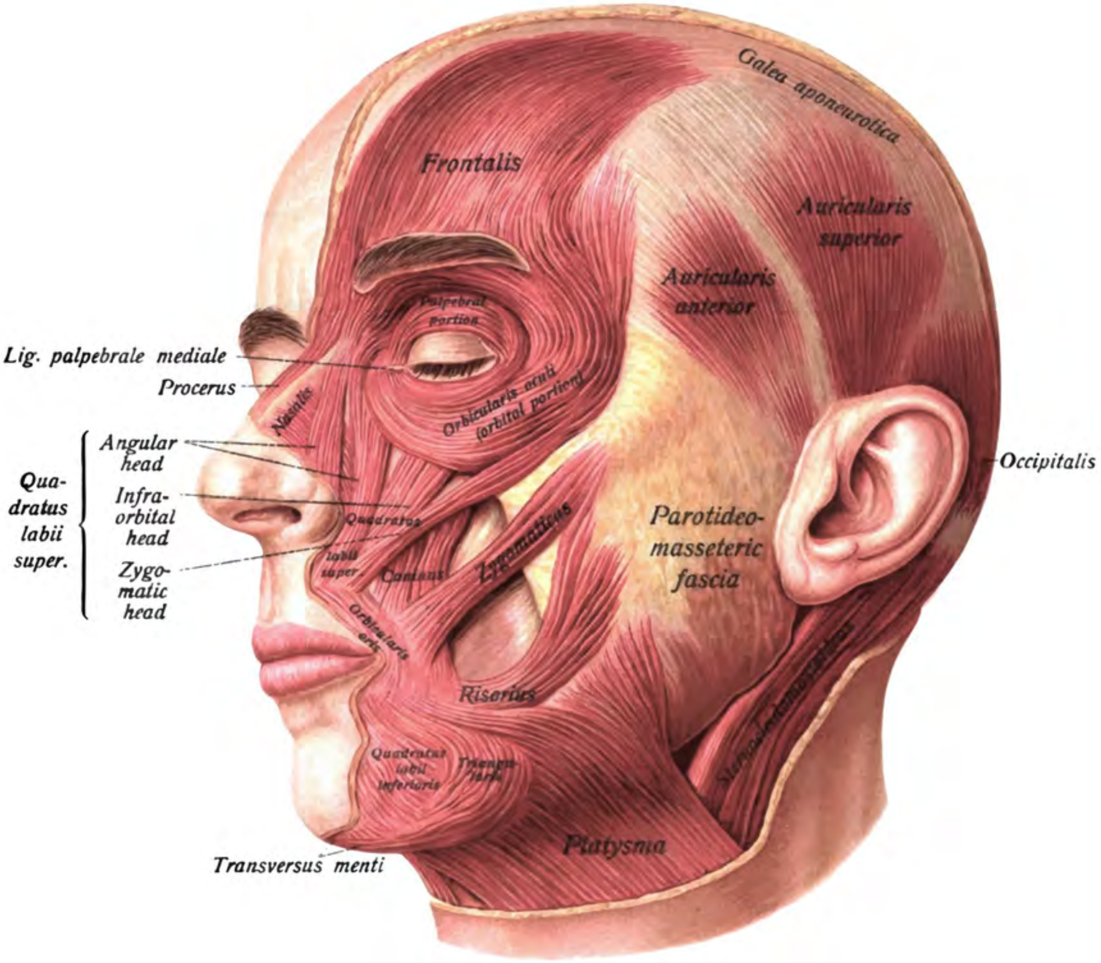


Figure 2.4: Facial muscles exposed on the left side of the face, neck, and head. This image was adapted from Sobotta's Atlas and Text-book of Human Anatomy [28].

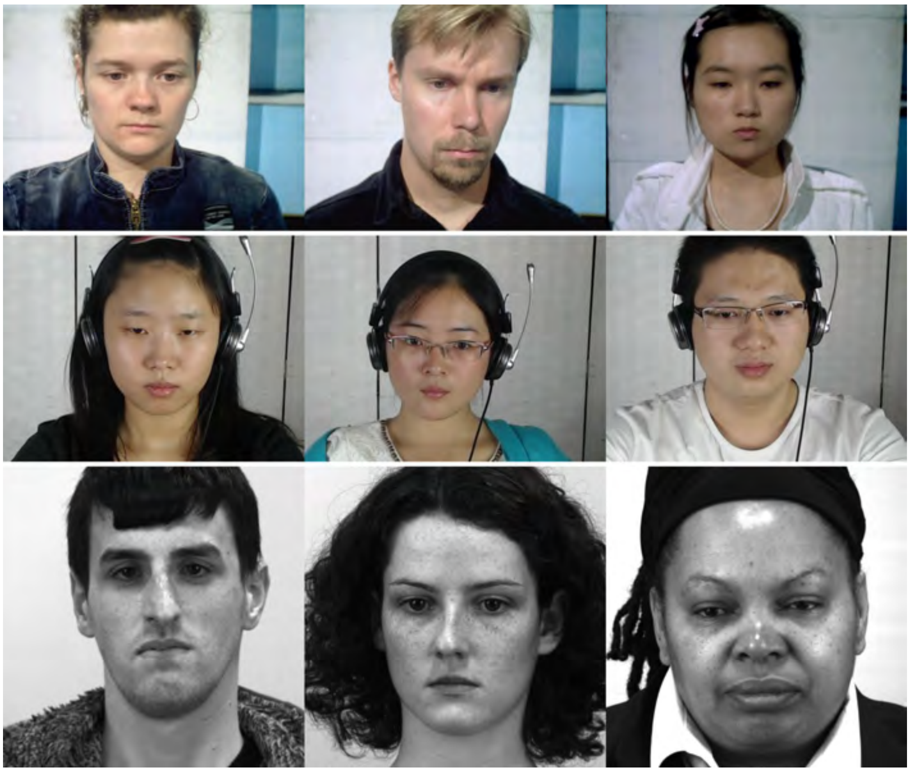


Figure 2.5: ME datasets. First row: SMIC [29], second row: CAS(ME)^2 [30], and third row: SAMM-LV [31].

deceptively describe two film clips that were either classed as emotional (i.e. surgical operations), or non-emotional (i.e. Hawaiian beach scene). The limitations of this paper are the lack of visible MEs in the dataset due to low frame rate and resolution. The authors suggest the encoders might be affected by camera shyness, and the motivation of participants to generate high-stake lies. It is a posed MEs dataset which makes it unnatural for analysis. This dataset is not publicly available.

USF-HD: Shreve et al. [34] demonstrated the first method to spot both MaEs and MEs. This dataset includes 56 videos containing 181 MaEs and 100 posed MEs with a frame rate of 29.7 fps and a resolution of 720×1280. The experiment procedure involved showing various MEs to participants and telling them to replicate those expressions in any order. Only 4 emotion classes were included in the dataset: anger, sad, smile, and surprise. The limitations are it is a posed MEs dataset and the videos were not FACS coded.

SMIC: Spontaneous Micro-expression Corpus (SMIC) [29] dataset is a ME dataset captured using different modalities. These include high-speed visual (SMIC-HS), normal-speed (SMIC-VIS) visual, and near infrared (SMIC-NIR). The experiment was conducted in an indoor bunker illuminated with four lights in the upper corners of the room. Twenty participants took part in this experiment, with a mean age of 26.7 years old (from 22 to 34 years) and are ethnically diverse. Participants sat in front of a computer monitor and were asked to watch 16 movie clips meant to induce strong emotions. For the recording process, the first ten participants were recorded using a high-speed camera with a resolution of 640×480 and a frame rate of 100 fps. Participants were asked to conceal their true feelings while watching the video clips and when they failed to do so, they would have to fill in a long questionnaire of more than 500 questions. The main limitation of SMIC is the assignment of each emotion into 3 relatively broad categories (positive, negative, and surprise). This contrasts with the seven basic emotion categories. The emotion inducing video clips varied in length from 9 seconds to 6 minutes, the lighting used was normal indoor lighting, which was occasionally affected by light flickering. There was also a gender imbalance in the participants, with 70 % being male.

CASME: Chinese Academy of Sciences Micro-Expressions (CASME) [35] contains 195 MEs with a frame rate of 60 fps. There are 35 participants (13 females and 22 males), with a mean age of 22.03 years with a standard deviation of 1.6. The samples were divided into two classes named Class A and Class B. Class A samples were recorded at a frame rate of 60 fps and resolution of 1280×720 pixels. The participants were recorded in natural light. Class B samples were recorded by a camera with a frame rate of 60 fps and resolution of 640×480 pixels. The participants were recorded in a room illuminated by two LED lights. During data acquisition, participants were expected to experience emotional high arousal and strong motivation to disguise their true emotions. This was done by instructing the participants to watch videos that elicit either highly positive or negative emotions while maintaining a neutral face at the same time. After the experiment, the participants were also asked to watch their own facial movements in the recordings and indicate whether they produced irrelevant facial movements which could be excluded from the analysis. The limitations of this dataset are some MaEs that were similar to MEs were included and assumed to be MEs in the dataset. The age of participants was in the same age group with a mean age of 22 years old (standard deviation of 1.6 years), and all participants also had the same ethnic background.

CASME II: Chinese Academy of Sciences Micro-Expressions II (CASME II) [36] contains 247 clips with a higher temporal resolution, and the size of the faces captured is larger than the previous datasets. The experiment involved 35 participants with a mean age of 22.03 years (standard deviation of 1.6). 18 participants were asked to maintain a neutral facial expression when watching high emotional valence video clips while 17 participants tried only to suppress the facial movement when they realised there was a facial movement. This dataset was captured by using a camera with a resolution of 640×480 pixels and a frame rate of 200 fps. In order to prevent light flickering, four LED lamps were used under umbrella reflectors to provide steady and high-intensity illumination throughout the data acquisition process. The limitations of this dataset are similar to the previous dataset [35], whereby the age of the participants were in the same age group of 22 years old and a standard deviation of 1.6 years and all of the participants also have the same ethnic background.

SAMM: Spontaneous Actions and Micro-movements (SAMM) [37] dataset has the highest resolution of all micro-expression datasets and the largest variation in participant ethnicity. Participants consisted of 32 people, with a mean age of 33.24 years (standard deviation of 11.32; aged between 19 and 57), from a wide range of ethnic backgrounds and an even gender split (16 male and 16 female). The dataset has a resolution of 2040×1088 and a frame rate of 200 fps. The data acquisition protocol was personalised to each participant instead of using generalised emotional inducement videos, as with previous datasets. The experiment was designed to detect the basic 7 emotions: contempt, disgust, fear, anger, sadness, happiness, and surprise. The participants were asked to suppress their emotions and keep a neutral face while watching the emotional inducement videos. The dataset is limited in its ability to generalise to real-world scenarios and the facial resolution captured is smaller relative to the overall image resolution.

MEVIEW: MEVIEW (Micro-Expression VIdEos in the Wild) [6] consists of videos from poker games and TV interviews downloaded from the Internet. There are 31 videos with 16 individuals in the dataset, with an average video length of 3 seconds. Poker games are a high-stress activity and players often need to hide their emotions, resulting in the potential for MEs. FACS coding and the emotion types are annotated. The limitations are that apex frames are not annotated and the sample size is small compared to other ME datasets.

2.3.2 Long Video Datasets

Long video datasets are a more recent addition to micro-expression research and resemble data closer to real-world situations. The motivation for creating these datasets is that MEs and MaEs can occur independently while overlapping in real-world scenarios. These datasets are useful for ME spotting tasks as it is more challenging. They contain more non-ME movements, which makes it more difficult for a machine learning model to distinguish between ME and non-ME movements.

CAS(ME)^2: CAS(ME)^2 [30] was the first publicly available dataset containing micro-facial expressions in long videos primarily used for micro-expression spotting. There are 22 participants with a mean age of 22.59 years (standard deviation of 2.2 years), and nine emotional videos are selected out of 20 videos as the emotion eliciting materials. Each emotion video is assumed to predominantly elicit one type of emotion. During the elicitation process, participants were seated in a room illuminated by two LED lights in front of a webcam (visible light camera) with a resolution of 640×480 and a frame rate of 30 fps. Nine videos are presented in random order and the participants were asked to maintain a neutral expression while watching them. The limitations of this dataset included assigning into four broad categories (positive, negative, surprise, and others) instead of the 7 basic emotions, with the 'others' class not being clearly defined. Participants were recruited of similar ages and were captured with a low frame rate of 30 fps.

SAMM-LV: SAMM Long Videos [31] is a ME dataset that contains long videos primarily made for ME spotting. The motivation for producing this dataset is the limited number of available datasets for spontaneous ME and macro-expression recognition and spotting. It is an extension of SAMM [37], with the data being captured during these experiments. The dataset consists of 147 long videos with 343 macro-expressions and 159 MEs. It was captured at a frame rate of 200 fps. Each video is FACS-coded with detailed AUs. The onset, apex and offset frames for every expression are also annotated.

SMIC-E-Long: SMIC-E-Long [38] is an extension of the SMIC dataset by appending neutral frames before and after the ME. In addition to the new dataset, a standardised ME spotting evaluation method (a frame-based evaluation which takes into consideration of mean deviation of the correct predictions) was proposed. The author claimed that most datasets append neutral frames before and after ME samples to produce longer videos. This dataset was created by adding more than 2000 to 3000 frames (about 20 seconds) before and after ME samples. This dataset also contains clips of regular facial expressions in some of the videos. There are a total of 162 long videos with an average of 350000 frames. The videos, with a resolution of 640×480, were captured using 16 different participants.

MMEW: Micro-and-macro Expression Warehouse (MMEW) [39] shared the same largest facial resolution with SAMM till date according to the author. This dataset contains 900 MaE, the largest in a ME-based dataset. Participants were asked to maintain a neutral expression while watching emotional inducement videos similar to previous datasets.

2.3.3 Summary of ME Datasets

Table 2.3: Summary of ME datasets. Ethnic.= Ethnicities

Year	**Dataset**	**Participants**	**Resolution**	**FPS**	**Samples**	**Emotion Classes**	**FACS Coded**	**Ethnic.**
2009	Polikovsky	11	640×480	200	13	7	Yes	3
	YorkDDT	9	320×240	25	18	N/A	No	N/A
2011	USF-HD	N/A	720×1280	29.7	100	4	No	N/A
2013	CASME	35	640×480, 1280×720	60	195	7	Yes	1
	SMIC	20	640×480	100 & 25	164	3	No	3
2014	CASME II	35	640×480	200	247	5	Yes	1
2016	SAMM	32	2040×1088	200	159	7	Yes	13
2017	MEVIEW	16	1280×720	25	40	7	Yes	N/A
2018	CAS(ME)^2	22	640×480	30	250 macro, 53 micro	4	No	1
2019	SAMM Long Videos	30	2040×1088	200	147	N/A	Yes	13
2021	SMIC-E-Long	16	640×480	100	167	3	No	3
	MMEW	36	1920×1080	90	300	7	Yes	1

Emotions are complex phenomena experienced differently by each individual. Producing emotional expressions, as in the non-spontaneous datasets, might not be viable as they are posed and do not resemble true to life expressions. Spontaneous datasets are a better alternative as the displayed emotions are as natural as possible in a lab-controlled environment. The current dataset pool for ME research is relatively small, with the need for more data being a top priority. Datasets produced must have high resolution, high frame rate, and be filmed within a consistent environment and lighting. Participants of the data acquisition should be recruited from a large demographic where possible to have a representative population. A better classification must be determined in order to obtain a better overview of the purpose of eliciting emotions. Another alternative to increase the samples within datasets is to use generative models to generate unique

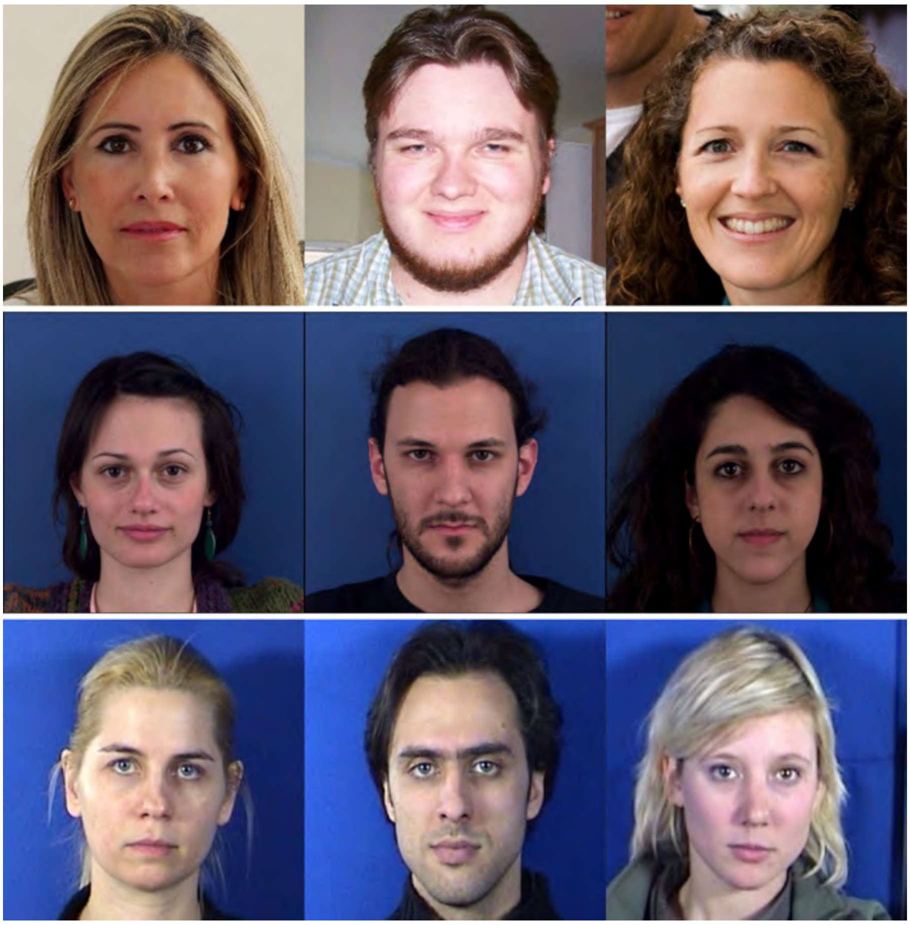


Figure 2.6: Face-related datasets. First row: FFHQ, second row: MMI (cropped to remove excessive background), and third row: the MUG dataset. FFHQ consists of face images; MMI and MUG are facial expression sequence datasets.

MEs.

2.4 Other Datasets

There are other face-related datasets used in this thesis. The visual examples is shown in Figure 2.6.

FFHQ: Flickr-Faces-HQ Dataset (FFHQ) [40] is a high quality human faces dataset. This dataset contains 70,000 PNG images with a resolution of 1024×1024. The author claims the dataset to have diverse ages, ethnicities, and image backgrounds. Facial accessories such as eyeglasses, sunglasses, and hats are also present.

MUG: The Multimedia Understanding Group Facial Expression Database (MUG) [41] is a facial expression sequence dataset split into 2 parts: posed expressions and spontaneous expressions. All image sequences are frontal face recordings. The first part contains 86 participants performing the six basic expressions (anger, disgust, fear, happiness, sadness, and surprise). Neutral face sequences of each subject were also captured. The second part contains the same subjects recorded while watching emotion-eliciting videos. All the participants were Caucasian and aged between 20 and 35 years. There were 35 females and 51 males. Each video has 50 to 160 image sequences. Fear was the most difficult expression to pose or induce and contained the most errors. The frame rate of this dataset is 19 fps. The image resolution is 896×896 pixels. The version used in our experiments consists of 52 subjects with annotations and is available online.

MMI: Induced Disgust, Happiness and Surprise: an Addition to the MMI Facial Expression Database (MMI) [42] is a facial expression dataset using audio-visual recording simulation. The first set of data involves participants posing all 31 AUs and a number of Action Descriptors. The participants are also asked to show the facial expression of two or three affective states (e.g. sleepy, happy, bored). A 45-degree mirror was placed next to the participants to record the side view of the participants. The second set consisted of posed displays of 6 basic emotions. Participants with glasses are told to record once with and once without glasses. The main purpose of this set is to record high-resolution facial expressions. The third set contains high-quality still images. Similar to the first set, the participants were asked to display all AUs and 6 basic emotions.

2.5 ME Recognition

ME recognition involves the classification of ME videos into one of the emotion classes. The common approach involves feature extraction followed by categorising them based on those features. Commonly used features are 3D Histogram of Oriented Gradients (3DHOG), Local Binary Pattern (LBP), and Optical Flow (OF).

2.5.1 Conventional Approaches

Conventional approaches are methods that use traditional handcrafted features and perform manual classification based on these features.

3D Histogram of Oriented Gradients (3DHOG) [43] evaluates well-normalised local histograms of image gradient orientation in a dense grid. It is characterised by local object appearance and shape by the distribution of local intensity of gradients or edge directions. This method divides the image window into small cells that accumulate a local 1-D histogram of gradient directions or edge orientations. The combination of local histograms is then accumulated over a larger spatial region ("blocks") and the cells in the blocks are then normalised. This feature is used in facial alignment and detection. There are several ME recognition methods [32], [44], [45] that use this feature as input.

Early ME recognition methods use Local Binary Patterns (LBP) [46], [47]. LBP compares the grey level value of the centre pixel with the neighbouring pixels. If the grey level of the neighbouring pixel is higher than the centre pixel, it is assigned to 1, while it is assigned to 0 otherwise (further details in Section 3.3). The most common variant used is Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) [48] in many ME recognition methods [29], [49]–[51]. This feature is an extension of LBP, where it takes in the local spatio-temporal neighbourhoods across three planes: spatial (XY), horizontal spatio-temporal (XT), vertical spatio-temporal (YT).

Optical flow (OF) [52], [53] is the pattern of apparent motion of image objects between two consecutive frames (further details in Section 3.3). Optical flow is divided into two types: sparse and dense optical flow. Sparse OF selects a set of pixels on sparse features (e.g. corners or edges) to track its velocity vectors; dense OF compute the optical flow vector on a per-pixel basis. There are some conventional ME recognition methods [51], [54]–[57] that use OF as input feature. Histogram of Oriented Optical Flow (HOOF) and Main Directional Mean Optical-flow (MDMO) are also used as input features [58]. This method uses Flownet 2.0 [59] for optical flow extraction and computed using revised HOOF features. The features extracted are then used for spotting MEs. However, only detection accuracy is reported.

Buhari et al. [60] show ME recognition can be conducted using combined features of the landmark-based facial graph, euclidean distance, and gradient of the graph segments. This method introduces an invisible emotion magnification using facial landmark points of the onset and apex frames. The performance evaluation is conducted using the magnification technique with landmark-based graph features. This method computes the landmarks of onset and apex frames using Histogram of Oriented Gradients (HOG) features combined with a linear classifier.

Emotion magnification is performed by calculating the geometrical distance between onset and apex frame. A magnification factor is used to adjust the emotion magnification level to an optimal level. The resultant magnified vector is the product between the magnification factor and facial muscle direction. This method uses the graph extracted from the face (full-face or FACS-based). It is also tested on the extracted Delaunay triangulation. The feature extraction-based method adds computational complexity due to the pre-processing steps. The apex frame of the face sequence was placed in the middle of the video. This method is a version of apex recognition, which is a limitation as the peak of expression needs to be located manually.

Pre-processing steps are often used to enrich the input features. Temporal Interpolation Model (TIM), used in pre-processing [61]–[63], is conducted to either increase or decrease the number of frames to produce videos of a similar frame rate. Motion magnification using Eulerian Video Magnification (EVM) [64] is used for directional amplification of movements by tuning the temporal filter to the desired frequency band. Using Laplacian transformation, the frequency domain of the ME sequences is obtained. Several methods [65]–[69] use EVM to amplify the facial movement as pre-processing.

2.5.2 Hybrid Approaches

In the advent of deep learning, recent ME methods use neural networks for analysis. However, most methods still use handcrafted features as input that is passed into artificial neural networks. We define these methods as hybrid approaches. Most methods still use inputs consisting of extracted features such as LBP, OF, and spatio-temporal features.

Methods that use LBP as input features are performed by fixed grid-based spatial division and an ROI method [70] or a combination of both handcrafted (LBP-based) and deep (CNN) features [71]. Both of these methods pass LBP-based spatio-temporal descriptors, LBP-TOP, with a selection of certain regions/blocks through a CNN. The feature or block size is a hyper-parameter that needs to be selected manually.

Most available methods use optical flow as input features. A few different types of OF used are Dense OF (i.e. Horn-Schunk, Farneback), Sparse OF (i.e. Lucas-Kanade), and TV-L1 optical flow. The further details of these features can be found in Section 3.3. OF is often fed into convolutional neural network with multiple streams (dual streams [72]–[75] and three streams [5], [63], [76]). The features used in each stream are usually OF (horizontal and vertical optical flow field) and optical strain (which is the derivative of optical flow that can estimate the intensity of facial deformation). There exist methods that pass OF into a recurrent network [67] and a hybrid between recurrent convolutional network [69]. Apex frame recognition using OF [66], [77], [78] shows great recognition accuracy, as the apex frame is the peak of the movement and yields the most distinct differences while comparing with the whole sequence. Although it shows higher accuracy, in real life, the apex frame is unknown and hence, limits this method to controlled environments.

Belaiche et al. [79] show that a deeper network does not necessarily result in much improvement in ME recognition. This method uses optical flow as input. This method mainly optimises the depth of the network to maximise the efficiency of the model. It also investigates the most suitable optical flow orientation and combinations. Authors report to have potential real-time performance but did not take into account the processing costs of optical flow. Furthermore, the accuracy reported is 0.6017, which is lower than most of the state-of-the-art models.

LEARNet [80] is designed to learn minute features of MEs using accretion layers. These features are processed spatio-temporal information of video sequences into one frame instance. These layers utilise hybrid responses generated by previous layers to enhance the learnability of minute features and maintain the feature maps simultaneously. It uses a small filter size (1×1, 3×3, 5×5) and does not contain a max pooling layer, instead using a convolution layer with a stride of 2 to downsample the feature maps. Dynamic imaging is used to combine spatial and temporal features into a single image. The main consideration for this design is to enable minute features to be detected more effectively.

2.5.3 Machine Learning Approaches

Machine learning approaches in this thesis are defined as artificial neural networks that input an image or image sequence without any feature extraction. Often with image input additional, pre-processing is needed, such as ROI extraction [81]–[83].

Most machine learning approaches use a CNN architecture with another auxiliary network. Spatio-temporal based approaches [62], [81] often use CNNs alongside a recurrent neural network (RNN), most commonly a long short-term memory (LSTM) network. Pure LSTM methods also exist [84]. The motivation of this network is to speed up computational efficiency and explore the interpretability and potential of deep neural networks by using Sparse LSTM. The pre-processing steps (facial area acquisition, data augmentation, grey-scaling, sequence length, and image size normalisation) are used and the features of the sequences are passed through LSTM units for classification.

Randomised frame selection as used by Xia et al. [68] while training can introduce more variety of input jitters to the system. This method performs ROI selection using an Active Shape Model (ASM) [85]. Eulerian Video Magnification [64] was used to enhance the subtle changes of MEs based on the aligned facial regions. The randomised frame selection is performed by randomly dropping frames based on a certain percentage. However, the main limitation is the processed sequence is always within 30 frames, whereby resizing methods is used to pre-process the input.

Transfer learning is also used by conducting training on a MaE dataset [62], [83], [86] before validating on a ME dataset. Most of the methods are pretrained on MaE, only Peng et al. [86] use pretrained weights on ImageNet before training them with the MaE dataset. However, MaEs and MEs are distinctly different in intensity and movement. Although it shows some improvement, it does not justify the viability of transferring movements learnt from MaE. These methods still do not outperform the state-of-the-art that uses only ME as the training set.

The CapsuleNet [87] architecture is also used in ME recognition (in apex frame recognition) by Van et al. [88]. This method is rotationally invariant and addresses the limitation of CNNs when handling inputs with different rotations. Van et al. use a CNN for feature extraction before the primary capsule. Authors compare VGG11 and ResNet18 and found that ResNet18 performs better. This can be explained as VGG11 utilises max-pooling more frequently and hence more spatial information of the features is lost, providing evidence to avoid max-pooling when using CNNs for feature extraction with CapsuleNet.

2.6 ME and MaE Spotting

ME and MaE spotting are a more recent development and research task. The definition of spotting is the detection of the presence of ME or MaE in a sequence with certain confidence (explained in Section 3.5.3). This task is proposed as the majority of ME recognition methods preemptively select apex frames from the image sequence using the ground truth. Both ME and MaE are used in this task as it closely resembles a real situation where they can occur simultaneously.

2.6.1 Conventional Approaches

Similar to ME recognition, conventional approaches are trained and/or evaluated on short video datasets. These approaches use traditional handcrafted features as input and perform manual facial movement spotting.

The first available ME datasets all consist of short videos from SMIC [29], CASME [35], CASME II [36], and SAMM [37]. Evaluating a few seconds of short ME clips will have a higher detection rate, as each video contains at least one ME. It does not resemble a real-world situation, where ME occurrence is rare and does not happen every few seconds.

The early works [44] of ME spotting are treated as classification tasks, where the frame sequences are categorised as neutral (where facial expression is absent), onset (where facial expression starts), apex (where facial expression has the highest intensity), or offset (where facial expression ends). Moreover, apex frame spotting [54], [89]–[91] is often incorrectly defined as ME and/or MaE spotting in the literature. Although apex frame spotting shows better performance, this does not imply a better model. Apex spotting is a less challenging task as the apex frame has the highest movement intensity within a facial expression sequence compared to spotting ME and/or MaE, which is an interval-based task. Similar to ME recognition, the input features used in spotting consisted of 3D-HOG [92], [93], LBP [54], [94], [95], and OF [96], [97]. For more recent methods, long video datasets from CAS(ME)^2 [30] and SAMM-LV [31] are used. These provide a more realistic scenario, whereby ME occurs alongside MaE and other facial movements.

2.6.2 Hybrid Approaches

Hybrid approaches are defined as artificial neural networks that use handcrafted features as input. The extracted input features include LBP, OF, facial action units, and spatio-temporal features. These methods are trained and/or evaluated on long video datasets containing both

ME and MaE. These datasets resemble real-world situations, where MEs are scarce, can co-occur and even overlap with MaEs.

LBP is used in apex spotting [98]. This method introduces Cubic-LBP to address the limitation of LBP-TOP which uses only features on three planes. Cubic-LBP has 6 cube faces and 9 planes intersecting with the centre pixel. The differences between frames are then calculated by using the sum of squared differences. This can further calculate the difference between two histograms. Other than a conventional CNN, recurrent neural networks, especially LSTM, are used in ME spotting. Tran et al. [99] use conventional features (LBP and HOG-based features) and feed them into an LSTM network. However, this method is apex frame spotting, which does not consider the evaluation as an interval-based task.

Optical flow remains the most used input feature for the spotting task. Zhang et al. [100] extract features using ROI segmentation and then obtain the OF of each respective ROI. HOOF is then calculated based on the OF detected. All the features detected are combined in a feature matrix and used for ME/MaE spotting. Yu et al. [101] use OF as input features and perform classification and regression simultaneously using a two-stream network. Verburg et al. [102] implement HOOF as input features and use LSTM to learn temporal information. This method uses post-processing, which suppresses the overlapping neighbours of a spotted interval. Sun et al. [103] is a two-stream network that uses images and optical flow as input features. The network architecture is a spatio-temporal cascaded network that consists of CNN and attention-aware LSTM.

Absolute frame difference is also used as input feature. Borza et al. [104] use only two absolute frame differences, but this method requires the face to remain static and the selection of the frame is conducted using half the interval of the expression (which is not possible in real life as we do not know exactly when the facial expression will end).

AUs extracted using OpenFace are used as input features [105]. The network architecture is a CNN with different kernel sizes (SAMM-LV uses 5, 7, 9 while CAS(ME)^2 uses 3, 4, 5). It primarily extracts features in different time samples and emphasises both local and global features by manipulating the dimension of the output space.

2.6.3 Machine Learning Approaches

This section includes machine learning approaches that use artificial neural networks with images or image sequences as input. No feature extraction is used for the input, making them unique in the ME/MaE spotting task.

There are very few methods that directly use images [106] and ROIs of images [107] as input. Pan et al. [107] pass the entire facial area and local ROI of each frame of the full video into a two-stream CNN. Wang et al. [106] proposed a 2D-CNN for capturing spatial information and another 1D-CNN for temporal information. This method uses one module to extract spatio-temporal features using a CNN, followed by another regression module which determines the temporal boundary by classifying whether the detection is a ME. While the reduced dimensionality lowers model complexity, the model begins to overfit. Using a 1D-CNN, the network potentially oversimplifies the input features, as there is less information to learn when compared to a CNN of higher dimensions, which is prone to overfitting.

Chanti et al. [108] describe MEs as abnormal patterns, or ‘anomalies’, that diverged from expected normal facial behaviour patterns. The method uses temporal multi-scaling by sliding a 20-time-step temporal window along the video sequence. The optimal length hyperparameter is set through a process of fine-tuning. The spatio-temporal information is extracted and classified using a Recurrent Convolutional Auto-encoder.

Most methods previously discussed start with pre-processing (face alignment, face crop, ROI extraction, manual frame selection) and ends with post-processing (manual thresholding, smoothing, signal merging), which increases the complexity of the model and real-time performance. An end-to-end approach is not available in the current research state, often requiring pre-processing [107] or fusion of features (i.e. spatial plus temporal features using a different CNN [106]). Our approach [109] aims to achieve close to an end-to-end pipeline, and will be further explained in Chapter 5.

2.7 Facial Expression Generation Models

Deep learning is a powerful, but data-hungry, machine learning technique. Hence, there is an ever-increasing demand for data. ME datasets are relatively small in size compared to other mainstream datasets (e.g. facial expression datasets). An alternative that is often explored is generative data using a deep learning based approach.

2.7.1 Macro-Expression Generation

There are a few ways to generate new facial expressions, including generating Facial Action Units (AUs) or movements on a neutral face and facial motion transfer. GANimation [110] takes advantage of AUs and proposed a generative model by linearly interpolating two facial expressions. This method takes a static facial image and transfers the facial expression onto it using AUs. The prerequisite is to obtain the AU annotation (e.g. using OpenFace [4]) and train the network to associate each facial expression with its respective AU.

Liu et al. [111] generate facial expressions solely using autoencoders. However, the resulting outputs are very low resolution (36×36 pixels) and require optical flow as an additional input feature.

Fan et al. [112] use action control variables to manipulate facial expressions on an image containing a face. It is often conducted using linear action variables. This method is not realistic, in particular when the onset and offset of a facial expression are not linear.

There are attempts [113], [114] that control targeted face properties by changing the latent representation. Controlling the latent representation can be useful in generating novel images. These methods have the limitation of generating artifacts in between some intensity or overlapping of latent variables.

ExprGAN [115] is a GAN that changes facial expression intensity continuously. This method separates the identity and expression representation. It is a conditional GAN that uses an expression control module to map an expression code to describe the expression intensity. However, this method only works on aligned and cropped faces.

AffineGAN [116] generates facial expressions by mapping a latent representation with the facial expression intensity of a certain region. This method uses "inverse transformation" to map an affine transformation to facial expression intensity from videos automatically without any manual annotation. The generated results show a performance of more than 50% when evaluating if Amazon Mechanical Turk (AMT) workers could be fooled. The limitation of this method, similar to other methods, is that it is generated on a per-image basis.

Siarohin et al. [117] show that video sequence generation can be done using a single image using key points and local affine transformation. It is conducted through decoupling appearance and motion information using self-supervised learning without any annotation or information on the object to be animated. This method uses a source image and a series of driving video frames as inputs. Authors use backward optical flow as the back-wrapping is differentiable using bilinear sampling. Keypoints were also used to conduct local affine transformation, which performs motion transfer as keypoint displacement.

Video generation GANs are also capable of generating facial expressions. Vondrick et al. [118] use spatio-temporal convolution in both the generator and discriminator. However, this method has an upper limit of approximately 1-second duration in the videos generated as its generator is trained wholly with 32-frame videos.

Saito et al. [119] proposed an end-to-end model that uses a temporal generator and an image generator. MoCoGAN [120] generates videos by mapping random vector sequences to video frame sequences. It retains the image feature content while the motion part is treated as a stochastic process. These methods are able to generate video sequences with facial expressions, however, the generated images are low-resolution (e.g. 64×64). MEs are typically very subtle motions, thus excessive down-sampling risks discarding the relevant features.

2.7.2 Micro-expressions Generation

ME-based data generation is still in its early stages. The first ME generation competition, Micro-expression Grand Challenge 2021 (MEGC2021) [121], tasked participants to transfer the motion of pre-existing ME clips onto another face. All participants of this challenge use pre-existing and a pre-trained network as the backbone, such as first-order motion model based [122], GANimation-based [123], and VGG19-based [124].

AU Intensity Controllable Generative Adversarial Network (AU-ICGAN) [125] is a ME-related AU transfer. Similar to MEGC2021, this model uses both CASME II [36] and SAMM [37] datasets as references, and source frames by transferring AUs of CASME II onto SAMM participants and vice versa.

Identity-aware and Capsule-Enhanced Generative Adversarial Network (ICE-GAN) [126] is a standalone network without pre-trained weights that uses the CapsuleNet [87] architecture. The CapsuleNet discriminator is selected, as it is sensitive to geometrical encodings of relative position. None of the methods discussed in this subsection made the generated dataset publicly available.

Due to the subtlety of MEs, we applied a style transfer method [127] on an existing dataset, SAMM-LV, using the state-of-the-art generative adversarial network based style transfer method, StarGANv2 [128]. The further details are discussed in Chapter 6.

2.8 Summary

Most of the current ME recognition or spotting methods rely on using extracted features as input. The feature extraction process might discard important features that deep networks can make use of. Secondly, there is a lack of datasets, especially realistic datasets (i.e. long videos containing MEs). Most generation methods are on a per frame basis, and generation of the subtlety of MEs is still unsolved. Lastly, there is a lack of end-to-end approaches that can leverage deep learning to its full potential, where the majority of methods use conventional features as input. Hence, the contributions of this thesis are centred around bridging the gaps mentioned in this chapter, by proposing a new long videos dataset, an advanced augmentation approach to create more data, an end-to-end approach in ME/MaE spotting, and a spontaneous facial expression generation method.

Chapter 3

Theories and Techniques

3.1 Introduction

This chapter aims to provide background theories and methods that are used for thesis contributions. It contains preprocessing steps, feature descriptors used, machine learning-related theories, machine learning methods and performance metrics to quantify the results.

In this research area, most of the existing approaches use preprocessing which functions to remove excessive unrelated information. Some common procedures are facial landmark detection, facial alignment and facial action unit detection. OpenFace [3], [4], a facial behaviour analysis toolkit, is the main facial preprocessing and analysis software used throughout this thesis (further details in Section 3.2.3). The feature descriptors (e.g. optical flow) are simplified features extracted from images used in this thesis are also described. Majority of our methods in this thesis are based on machine learning especially convolutional neural network (CNN). Other than CNN, this chapter also explores alternative machine learning approaches including Restricted Boltzmann Machine, autoencoder and auto-regressor which are the backbone architecture of our sequence generation models. To quantify the performance of our methods, we primarily use interval based F1-score for spotting approach and image quality analysis is used for the generative models.

3.2 Preprocessing

Preprocessing is generally the first step of the input pipeline for image related tasks. This process selects wanted data by removing unwanted information. For face-related research, extracting regions of interest (which is the face) from an image is a common preprocessing step.

3.2.1 Face Alignment

Face registration is the geometrical alignment of the face based on facial features. It involves aligning two images together based on certain points of reference. The common method involves applying affine transformations (e.g. translation, rotation, scaling etc.) by taking reference on the stationary section of the face. A simple approach is to rotate 2D facial images using the centroid between both eyes as the center of rotation. In our approaches, we utilise facial alignment of OpenFace which is based on the detected facial landmark.

3.2.2 Facial Landmark Detection

Facial landmark detection is a process that determines the points on the face and tracks the movement of these points. Active Shape Model (ASM) [85] is a popular method due to its speed and robustness towards noise. It utilises a Point Distribution Model (PDM) [129] that rapidly locates the geometry of a certain shape depending on the training set used (shown in Figure 3.1). For the model structure of a human face, commonly 68 landmark points are implemented.

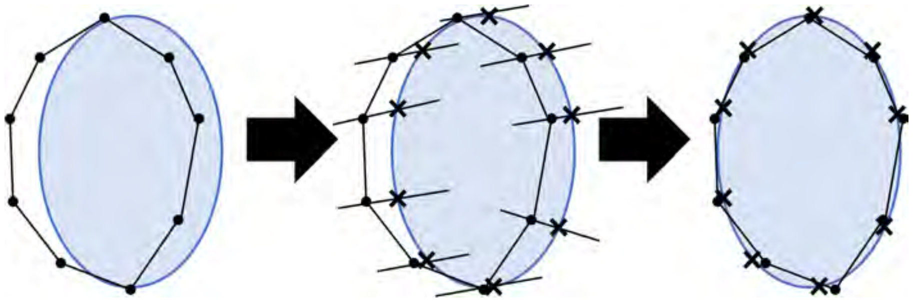


Figure 3.1: Segmentation step of Active Shape Model visualised. ASM attempts to match the mean shape model near the image object and the model parameters are modified to move the shape model to the best position (crosses on the shape). This figure was adapted from Pilch et al. [130].

OpenFace can either uses Constrained Local Model (CLM), Constrained Local Neural Field (CLNF) [131] or Convolutional Experts Constrained Local Model (CE-CLM) [132] for facial landmarking. Constrained Local Model (CLM) [133] is a more recent approach, which trains a patch around each feature and the patch is subsequently normalised. To apply this model on faces, it functions by learning to match a template that is constrained by facial joint shape and texture model built from labelled faces. CLNF is a probabilistic local landmark detector which can map the non-linear and spatial relationship between input pixels and aligned landmark. There are two spatial relationships captured: spatial similarity of nearby pixels and the entire area of the landmark detectors. This method is able to capture the continuous complex non-linear long and short-distance relationships between pixels. CE-CLM is an upgraded version of CLM with a novel local detector. This local detector has the advantage of a neural network and a mixture of experts. It has two main parts which are response map computation using experts of network and shape parameter using a PDM. It has the ability to model very complex individual landmark appearance that has different facial hair, illumination and expression. This method shows the best facial landmark detected among the three landmark approaches used by OpenFace.

A newer landmark approach is Style Aggregated Network for Facial Landmark Detection [134] which takes two images as its input where one of them is the original image while the other is a transformed image based on the original image. The transformed images are created by using PhotoShop and together with the original image, they are used together to fine-tune ResNet-152 which is the ImageNet pre-trained model. Next, by using k-means clustering, the hidden styles of the images are ”determined”. After that, all the clustered data are used to train the style transformation model via CycleGAN. Lastly, the facial landmark module is trained by initialising the first four convolutional blocks using VGG-16 ImageNet pre-trained model and the other layers are initialised using Gaussian distribution.

3.2.3 OpenFace

OpenFace [3], [4], [135] which is primarily used as a facial behaviour analysis toolkit. This toolkit is capable of facial landmark detection, head pose estimation, facial action unit recognition and eye-gaze estimation. For our research, we utilised it on face alignment and detection of AUs only.

OpenFace 2.0 was used in our research. The main advantages compared to the previous version are it uses an improved facial landmark detection system, distribution of ready-to-use trained models, increased real-time performance, cross-platform support and codes available in multiple programming languages.

For the facial alignment function, it uses Convolutional Experts Constrained Local Model (CE-CLM) [132] to detect and track facial landmarks. This model uses two models where it conducts response computation using Convolution Experts Network (CEN) and update shape parameter using Point Distribution Model (PDM). CE-CLM also adapt an 84-point landmark detection by using a fully connected deep neural network to map the output of CE-CLM and a second network to learn dataset-specific corrections for CE-CLM.

For facial expression recognition, the Facial Action Units (AUs) presence and intensity were conducted by using appearance features (Histograms of Oriented Gradients) and geometry features (shape parameters and landmark locations) [136]. The AUs able to be detected by OpenFace are shown in Table 3.1.

Table 3.1: List of AUs that can be detected using OpenFace

AU	**Full name**
AU1	Inner Brow Raiser
AU2	Outer Brow Raiser
AU4	Brow Lowerer
AU5	Upper Lid Raiser
AU6	Cheek Raiser
AU7	Lid Tightener
AU9	Nose Wrinkler
AU10	Upper Lip Raiser
AU12	Lip Corner Puller
AU14	Dimpler
AU15	Lip Corner Depressor
AU17	Chin Raiser
AU20	Lip Stretcher
AU23	Lip Tightener
AU25	Lips Part
AU26	Jaw Drop
AU28	Lip Suck
AU45	Blink

3.2.4 Local Contrast Normalisation

Local Contrast Normalisation (LCN) is used to normalise the contrast of an image by conducting local and divisive normalisation [137]. It performs normalisation on local patches (per pixel basis).

LCN operates by firstly defining a local region and subtracting each pixel with its local mean. Then, the variance of the region is calculated and if the variance is larger than 1, the target pixel is divided by that value.

Gaussian filters are used in our LCN implementations. Gaussian convolutions are used to obtain the local mean and variance. It is a low pass filter which reduces noises and also speeds up the local normalisation process as it is a separable filter (where 2-dimensional data can be calculated using 2 independent 1-dimensional functions).

The general equation of LCN is shown in equation 3.1. The results of LCN can be shown in Figures 3.3 and 3.2. LCN has good performance in normalising local variation in brightness and contrast [43] which makes it suited for correcting uneven illuminations or shading artifacts [138] as shown in Figure 3.2. The lower image was purposely made to have higher contrast. LCN is able to extract similar facial features as shown in both images on the right.

$$g(x,y) = \frac{f(x,y) - \mu_f(x,y)}{\sigma_f^2(x,y)} \tag{3.1}$$

where $f(x,y)$ is the input image, $\mu_f(x,y)$ is the local mean estimation, $\sigma_f^2(x,y)$ is the local variance estimation and $g(x,y)$ is the output image.

3.2.5 Temporal-based processing and Magnification Methods

Temporal Interpolation Method (TIM) [139] is used to address uneven time sequences. It is commonly used to normalise video length by adding neutral frames between frames that contain signals. For example, normalising downsampled ME video shown by Pfister et al. [140].

The video magnification method is used in magnifying the movement of faces. This method generally involves amplifying movements by exaggerating subtle colour and motion changes [64]. This primarily addresses the subtlety of ME, which is sometimes too faint to be registered as a signal for the algorithm to detect. Meaningful information such as raising eyebrows and lip tightening could be amplified making them more visible for the algorithm to identify. The examples of methods used are Eulerian Video Magnification (EVM) [64], DVMAG [141], Phase-based video magnification [142] and Riesz-based pyramids [143]. However, this method might

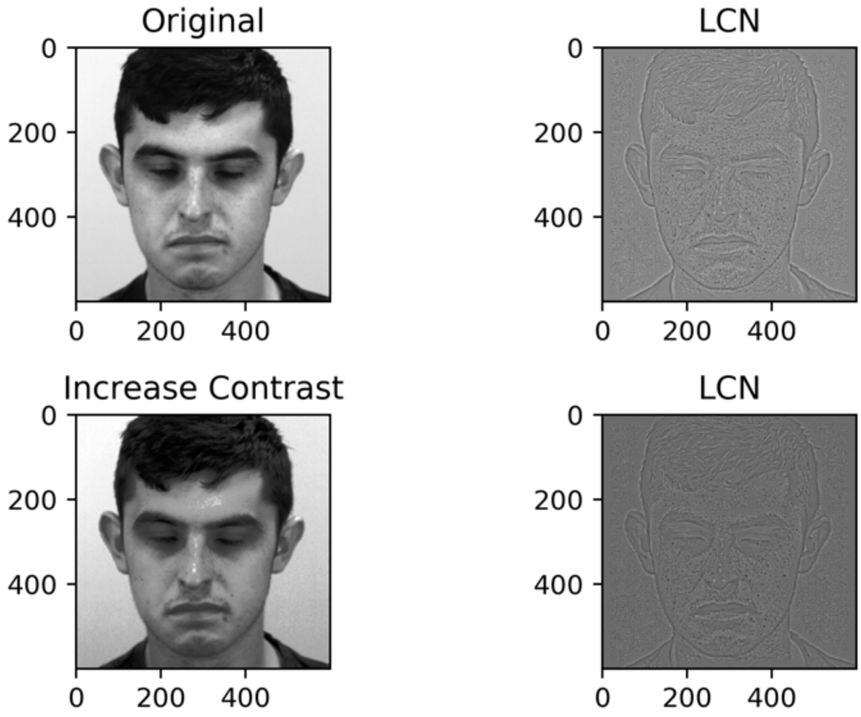


Figure 3.2: LCN on images with different contrast. The images on the right are results of LCN. The extracted features for both cases are visually similar.

also amplify movements that do not resemble any emotion, such as head movement and eye blinks which become noise to the learning algorithm.

3.3 Features Descriptor

Feature descriptors are algorithms used to extract features from raw data (raw images in computer vision). These features are relatively lower level than the raw images and processing these features requires lesser computational cost. The features can be used as a unique identifier for a certain object or motion and used in different analyses to quantify or qualify the model performance.

Local Binary Pattern (LBP) [46], [47] uses a window and compares central pixels with neighbouring pixels. If the grey level of the neighbouring pixel is higher than the centre pixel, it is assigned as 1 while it is assigned to 0 otherwise. There existed also advanced LBP-based features such as volume local binary patterns (VLBP) and Local Binary Pattern on Three Orthogonal Planes (LBP-TOP) [48]. VLBP is an extension of LBP using dynamic texture features as a

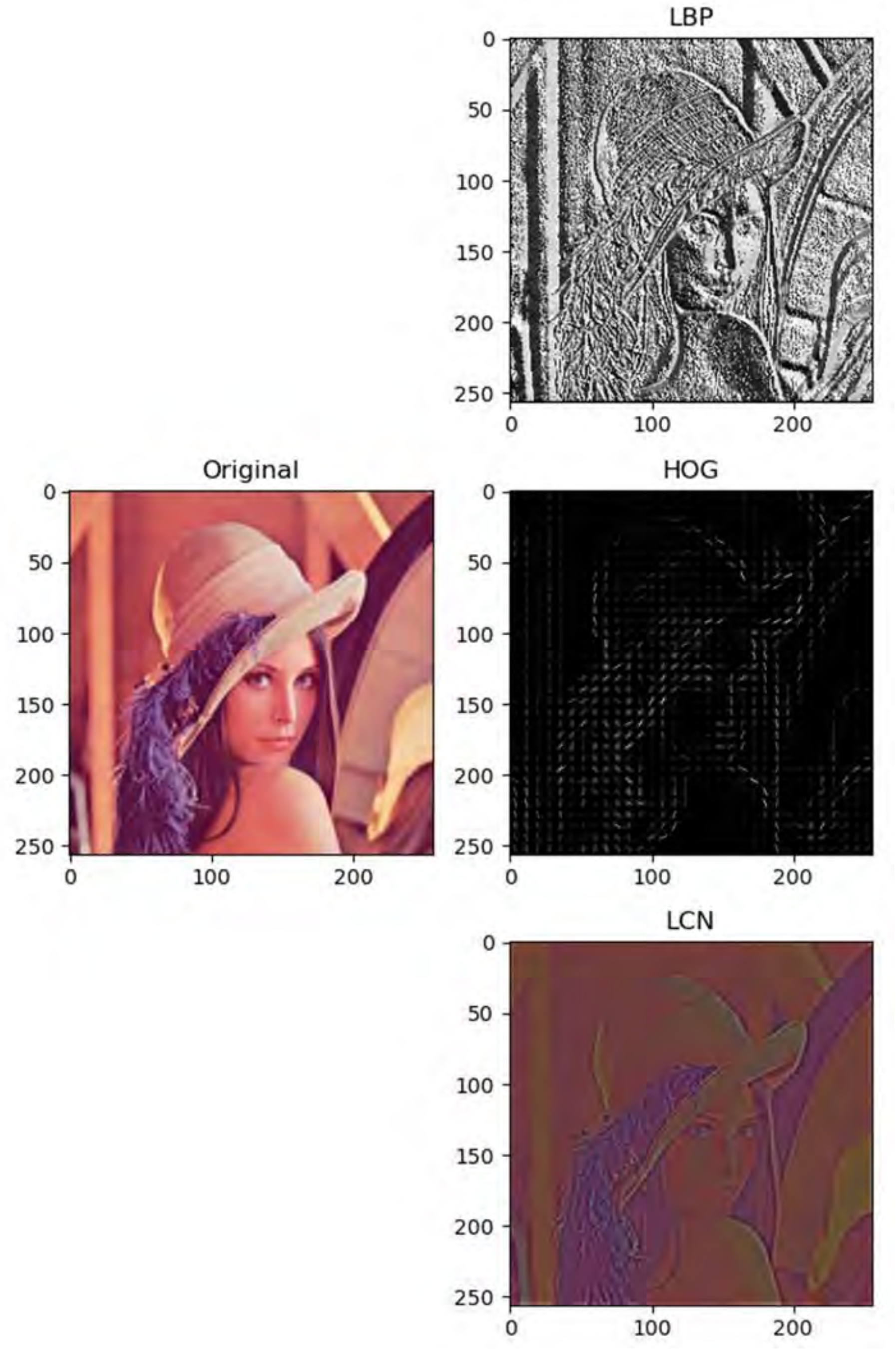


Figure 3.3: Comparison between different features extracted and normalisation method using the same input image. The first image is Local Binary Patterns (LBP), the second image is Histogram of Oriented Gradients (HOG) and the third image is processed using Local Contrast Normalisation (LCN). Our spotting method uses LCN to address the uneven brightness of the image by normalising the local contrast values of the image.

set of volumes of (X,Y,T) space where X and Y are spatial coordinates and T is the temporal information. It is computationally simple and easy to interpret. However, the main issue with VLBP is when the number of neighbouring points increases, the number of patterns of basic VLBP will become very large which limits its applicability. LBP-TOP concatenated the local spatio-temporal neighbourhoods across three planes: spatial (XY), horizontal spatio-temporal (XT), vertical spatio-temporal (YT). The difference between VLBP and LBP-TOP is VLBP uses three parallel planes which tends to make the feature vector too long while LBP-TOP uses three orthogonal planes which makes the feature vector much shorter when the number of neighbouring points increases. A visualisation of LBP can be seen in Figure 3.3.

Haar Features are digital image features which gets its name from Haar wavelet. Haar wavelet is a discrete signal with a time function that occurs at a discrete time interval. It is a sequence of square-shaped function. It is also used in the first real time face detector. These features are shown in Figure 3.4.

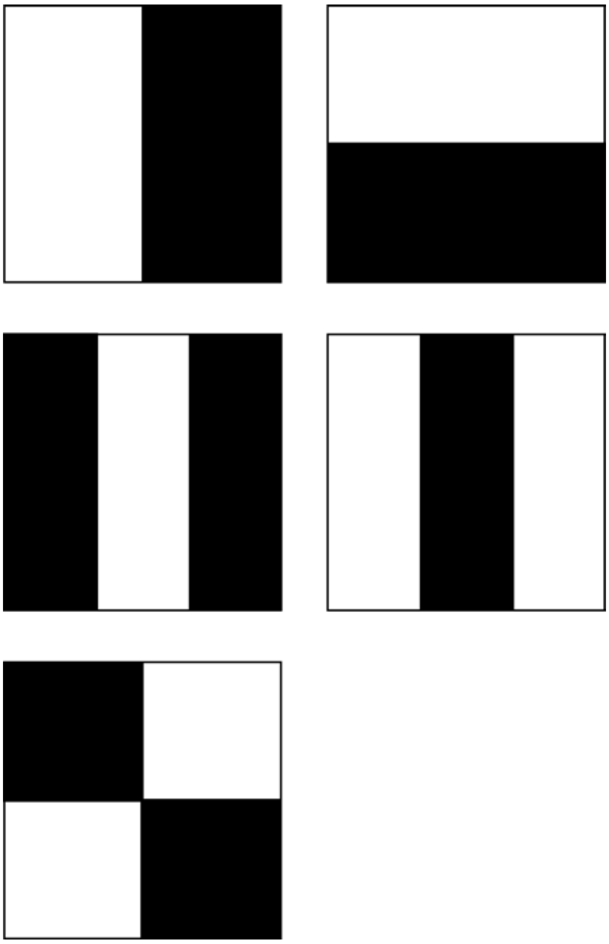


Figure 3.4: Haar Features are the simplest form of wavelet that consisted of a square-like function. The first row consists of edge features, the second row is line features and the third row is a four-rectangle feature.

Histogram of Oriented Gradients (HOG) [43] is primarily used in object detection. The basic idea is to divide images into certain regions called cells and merge the histogram of gradient directions calculated from these cells. The general steps involve computing gradients (horizontal and vertical gradients), binning the orientation of the histogram gradient of each cells, normalisation and calculate the HOG feature vector. The main advantage of this feature is it is photometric and geometric transformation invariance. A visualisation of HOG can be seen in Figure 3.3.

Optical Flow is the apparent motion of brightness patterns in the image [52]It is a popular feature used in ME research. This feature visualises apparent velocities of movement using brightness patterns in an image. It is divided into two main categories: dense optical flow and sparse optical flow. Dense optical flow takes into account all points while sparse optical flow processes only a part of the image. Horn-Schunck [52] is one of the early versions of optical flow. It is a dense optical flow method. It enforces global constraint of smoothness to address the aperture problem. This is done by minimising the global energy function. Farneback optical flow [144] uses polynomial expansion transform to estimate the quadratic polynomial of each sequence pair. It is a dense optical flow method. It uses an image pyramid where optical flow is computed from the lowest to the highest resolution of the image. This pyramid structure can track larger displacement. The advantage of this method is the computational cost is lower and the algorithm runs faster. Lucas-Kanade [53] has a different take on optical flow. It divides the image into smaller regions and computes a weighted least-square fit of the displacement between two frames. This method has an assumption that local optical flow is constant. It is a sparse optical flow method. There exists an enhanced version of this algorithm which uses image pyramidal representation [145]. TV-L1 Optical Flow Estimation [146], [147] estimates the plain intensity difference between pixels as an image similarity score. TV stands for total variation and L1 is a robust L1 normalisation. This method is separated into two modules where the first procedure performs calculations on the optical flow at a given scale numerically while the second implements pyramidal structure and approximates the extracted features.

Histogram of Oriented Optical Flow (HOOF) is an algorithm which sorts the extracted optical flows into bins and calculates the histogram distribution based on them. It addresses the susceptibility of optical flow towards background noise and scale changes of movement directions of the flow. The steps involved computing optical flow every frame, bin optical flow vectors into orientation bins (pixel angle and magnitude) and making histogram distributions based on the bins.

Facial Action Unit (AU) (further details in Section 2.2.1) is an organised annotation system which primarily based on muscle movements of the face. AUs can also be used as features and it is possible to be extracted automatically using computational approaches. OpenFace uses both appearance (HOGs) and geometry features (facial landmark position and shape parameters) to perform automatic real-time FACS detection [136]. Multi-View Dynamic Facial Action Unit Detection [148] performs AU detection using the optical flow field of video of human head to predict the viewpoint from which the video was taken. The main difference of this system when compared to the mainstream approaches is it directly analyses the information on the whole human face by bypassing landmark localisation.

Robust Principal Component Analysis (RPCA) is a statistical approach that decomposes the input data matrix into two parts: a low-rank subspace and an error term. In ME research, according to the author [149], irrelevant information such as pose and subject identity might overshadow important emotional information. The author uses the error term for RPCA for ME recognition. The use of RPCA will remove the identity of the subject (the RPCA low-rank subspace of the face) which is claimed to improve the detection accuracy.

3.4 Statistical and Machine Learning Method

Our methods in this thesis are based on machine learning and the basic idea originated from statistical approaches. The basic building block of our methods are based on CNN, Restricted Boltzmann Machine, autoencoder and auto-regressor.

3.4.1 Bayes' Theorem

Bayes' theorem is a method to calculate conditional probability based on previous events. It uses prior knowledge to predict the current state. Bayes' Theorem is the backbone for classification models in machine learning such as the Naive Bayes and Bayes Optimal Classifier. The general equation which describes Bayes' Theorem is as below:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \text{ and } P(B) \neq 0 \tag{3.2}$$

where A and B are events, P is the probability of certain events.

3.4.2 Markov Chain and Methods

Markov chain is a mathematical system that predicts the next state based only on the current state. It is often called a “memoryless” stochastic process.

A transition matrix for Markov chain is a matrix containing information on the probability of transitioning between states. As long as the sequence follows Markov property, the state of regular Markov chain will eventually converge after multiple multiplications with the transition matrix. A few examples of Markov-based models are Restricted Boltzmann Machine (RBM)

[150], [151] and Deep Belief Network [152]. The successive state matrices of regular Markov chains will always approach a stationary matrix. The general equation which describes Markov property is as below:

$$M(x_{n+1}|x_1, x_2, ..., x_n) = M(x_{n+1}|x_n) \tag{3.3}$$

where M is a function with Markov property, x is the input and n is the number of repetitions.

Markov chain Monte Carlo (MCMC) method is an algorithm that samples from a probability distribution using Markov chain that has desired distribution as its equilibrium distribution. This method is useful for the approximation of multi-dimensional integrals.

Markov-based and Bayesian model has very similar network representation. However, the difference is Bayesian model is directed and non-cyclic while Markov-based model is undirected and may be cyclic.

3.4.3 Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM) [153] is a two-layer neural network which is capable of learning the probability distributions of the inputs. This algorithm is trained using Contrastive Divergence. Contrastive Divergence estimates the gradient of energy functions by using many cycles of MCMC sampling which transform training data into a distribution. RBM has binary hidden and visible layers. The training of RBM involves minimising an energy function. The energy function involves the product of probability assigned to the training set as shown below:

$$E(v, h) = -\sum_{i \in visible} a_i v_i - \sum_{j \in visible} b_i h_i - \sum_{i,j} v_i h_j w_{ij} \tag{3.4}$$

where v_i, h_j are boolean visible (subscript i) and hidden (subscript j) states; a_i, b_j are each respective biases and w_{ij} are the weights between visible and hidden states.

The weights of RBM for reconstruction of MNIST digits is shown in Figure 3.5.

L1 and L2 regularisation are investigated. With L1 regularisation, the weights appear more contrast while L2 regularisation makes the weights blur. This makes L1 regularisation more suitable for tasks with larger differences in texture whereas L2 regularisation does not work well in samples containing outliers (due to the square difference that gives rise to larger errors). This is better for model convergence in general as similar features are grouped and make the learning more effective.

3.4.4 Moments

Moments of a function in mathematical definition describe the shape of the function. The first and second moments represent the mean and variance of a function. This is an organised notation to describe the detailed properties of a function. A prime example is Adam optimiser (details in Section C.6) which uses the first and second moment of the derivative of the loss function.

3.4.5 Autoencoder

Autoencoder is an unsupervised learning algorithm which learns efficient data encodings. The architecture is consisted of at least one encoder (which downsamples the input) and one decoder (which upsamples the latent representation). This network is trained by minimising the reconstruction error using gradient descent over the parameters of the networks. It is primarily used for dimension and complexity reduction. There are a few types of autoencoders such as sparse autoencoder, denoising autoencoder, deep autoencoder, sequence-to-sequence encoder and variational autoencoder. Applications of autoencoder include data compression, data denoising, anomaly detection, seq-to-seq prediction and image generation.

The majority of sequence generation using autoencoder is on tasks such as language processing or trend lines. Images or video sequence generation using pure autoencoder [154] is not common as generated images/videos are blurry due to information loss of the encoder downsampling.

Both RBM and autoencoder attempt to reconstruct the original input. The difference is RBM has only two layers of network while autoencoder has at least 3 layers (one act as bottleneck). RBM uses forward and backward passes on both visible and hidden layers while autoencoder only uses forward pass.

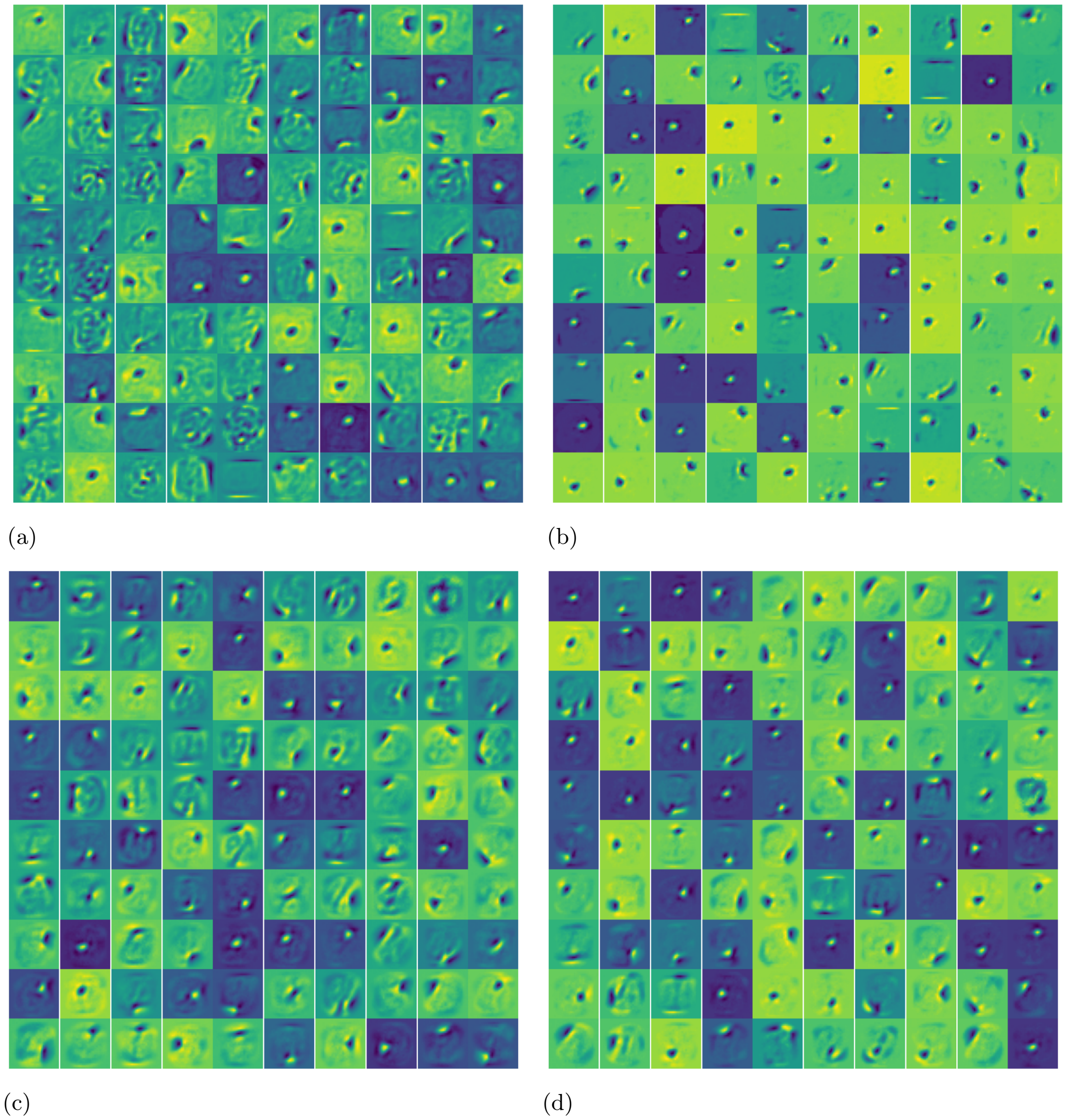
3.4.6 Variational Autoencoder

Variational Autoencoder (VAE) is an autoencoder commonly used for image generation. VAE is trained via constrained optimisation where it tries to minimise the reconstruction loss (between encoded-decoded data and the input data) and the KL divergence (between the data distribution and the model's marginal distribution) simultaneously. The difference between VAE compared to vanilla autoencoder is it samples around a mean latent representation within a certain variance, instead of a single value encoding. This "probabilistic" sampling nature (in contrast to "deterministic" sampling by regular autoencoder) of VAE enables randomised generation of the output. This is done by sampling through a Gaussian distribution with a mean within certain covariance.

The probabilistic sampling of VAE follows Equation 3.5 where z is the latent representation, N is the Gaussian Distribution, μ is the mean and δ is the covariance.

$$z \sim N(\mu_x, \delta_x) \tag{3.5}$$

The weakness of this generative model is the output is blurry as VAE learns the data distribution explicitly by attempting to fit the latent representation extracted into a multi-dimensional Gaussian distribution. The fitting of discrete latent representation on a continuous distribution spreads the probability mass diffusely over the latent space [155] that contains information loss which makes the output blurry. There are a few methods which attempt to address this issue such as Vector Quantised-Variational Autoencoder (VQ-VAE) [156]. Traditional VAE learns a continuous latent representation while VQ-VAE learns discrete latent representation. This discrete latent representation is generated by defining a certain size and dimension of discrete latent space. The nearest neighbour is calculated to determine whether the variable belongs within the defined latent space. By learning a discrete version of latent representation, posterior collapse (a condition in which the learned latent space becomes uninformative) which commonly affects VAE can be avoided.



(a)

(b)

(c)

(d)

Figure 3.5: The weights of RBM using different regularisation, where (a) RBM without any regularisation, (b) RBM with L1 regularisation, (c) RBM with L2 regularisation, (d) RBM with L1 and L2 regularisation.

3.4.7 Autoregressive Model

Auto-regressor [157] is a model that regresses on previous values of the time series. It is a time series algorithm that predicts the next time step based on information from previous time steps. Autoencoder can be used to predict the next element in a sequence in a similar manner by regressing on the previous latent representations. Generally, it can be thought of as the chain rule factorisation in a Bayesian network as shown in Equation 3.6.

$$[H]p(x) = \prod_{i=1}^{n} p(x_i|x_1, x_2, \ldots, x_{i-1}) = \prod_{i=1}^{n} p(x_i|X_{<i}) \tag{3.6}$$

where $X_{<i} = [x_1, x_2, \ldots, x_{i-1}]$ is random variable vectors with index less than i.

3.4.8 Convolutional Neural Network

Convolutional Neural Network (CNN) is commonly used in computer vision tasks. It is an artificial neural network with many local neurons with shared weights. 3D-CNN is a CNN with an extra depth dimension. The depth dimension allows more maneuverability of the network. It can be used to encode information spatially or temporally (we use depth as temporal information).

There are some unique network structures used in this thesis:

Depthwise Separable Convolution

Depthwise separable convolution inspired by MobileNet [158] has the ability to reduce computational cost. This operation is divided into two steps: depthwise and pointwise convolution. Depthwise convolution is a convolution that is applied on individual channels unlike the regular convolutional operation (which processes all channels at once). Pointwise convolution is convolution that uses a 1×1 kernel with the third dimension of c (where c is the number of channels) on the feature maps. The mechanism of depthwise separable convolution is shown in Figure 3.6.

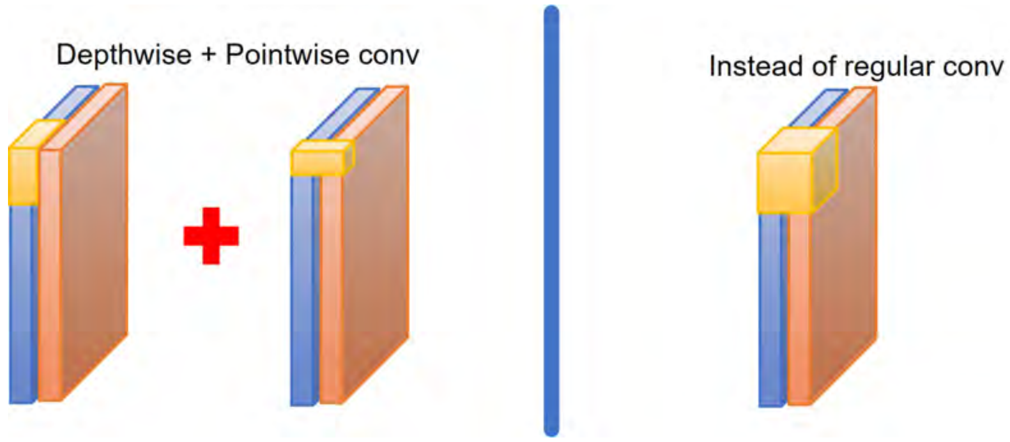


Figure 3.6: Depthwise separable convolution which is consisted of depthwise convolution followed by pointwise convolution. By conducting the channel-wise convolution separately, the computational cost is reduced as the number of parameters in this convolution is fewer.

By using these operations, it reduces the total number of parameters needed which addresses overfitting issues and it is also less computationally expensive. Standard convolutions have a computational cost of $D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$. The equivalent depthwise separable convolutions have a cost of $D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$. By comparing both costs, we observe a reduction in computational cost in using depthwise separable convolution as shown in Equation 3.7 with higher output channels and larger kernel size. (where M is the number of input channels, N is the number of output channels, $D_K \cdot D_K$ is the kernel size and $D_F \cdot D_F$ is the feature map size.)

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{3.7}$$

Residual Dense Layer

Residual dense layer is a ResNet [159] inspired dense layer. It consists of two or more dense layers that contain shortcut connections which perform identity mapping. This does not increase the number of parameters or computational complexity. The concept of a residual block is included Figure 3.7.

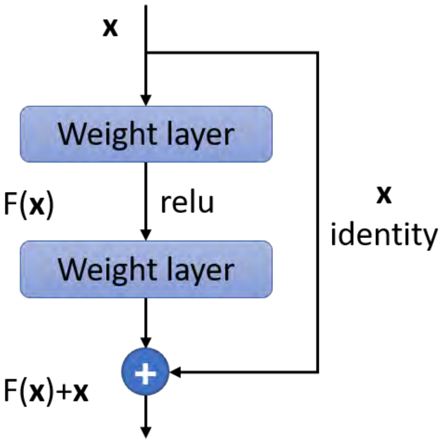


Figure 3.7: Residual Block that enables features to either go through or bypass the weight layer.

3.4.9 End-to-end Methods

End-to-end learning in machine learning is defined by the model that learn all the steps between the input and the output in a single pipeline. This type of model is more convenient compared to component-based models as end-to-end models can work on new tasks by changing input data. There are also fewer manual defined hyperparameters which makes the codebase to maintain easier. It also removes the dependency on other methods or feature extractions which reduces bias. A few approaches [160], [161] show action detection can be conducted in an end-to-end manner.

3.4.10 Multiple stream convolutional network

A conventional convolutional network uses single input and produces a single output. Multiple stream CNN (also known as multi-headed CNN) takes in two or more features. This CNN extracts features through different CNN (either with shared weight or not), concatenates those features at a certain part of the CNN and provides an output value. The simplified architecture is shown in Figure 3.8. There are different types of multiple stream CNN, they can be categorised into two major categories which are multi-stream CNN that uses different input features (such as CNN that uses optical flow and optical gray) and multi-stream CNN that uses the same input features (such as CNN that uses two images sampled from a different time sequence).

Multi-stream CNN with different input features: Two-stream convolutional neural networks by passing spatial information into one stream of network and temporal information (i.e. optical flow) into another separate stream of network [162], [163] are shown to be useful in action recognition tasks.

Multi-stream CNN with same input features: There are versions that uses the same feature such as using images of different time frame for each stream (slowfast network [164]), a hybrid of CNN+LSTM [165] and three stream variation also existed [166]. Inspired by action recognition task, multiple stream approaches are popular in ME tasks (ME recognition, ME and MaE spotting) shown in Chapter 2.

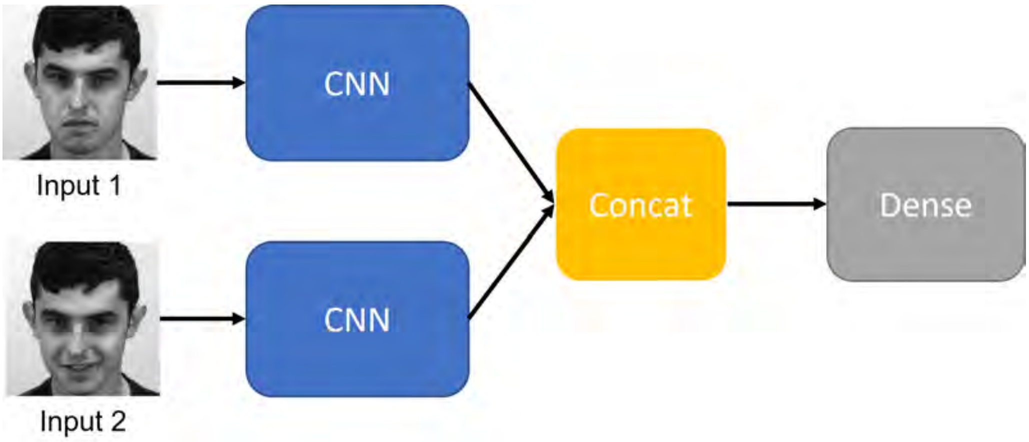


Figure 3.8: Simplified architecture of multi-stream CNN. This variation is the simplest version which takes in the same input features in two separate streams of CNN.

3.4.11 Neural Style Transfer

Neural style transfer is a method that involves two images which take the extracted style from an image and maps it to a source image using an artificial neural network. This method can increase data variation. The origin of this method is based on Gatys et al. [167] using different convolutional layers of a CNN. The generated images using this method do not look real; it looks visually similar to abstract art. A combination between CNN and Markov random field by Li et al. [168] is an improvement on Gatys et al. [167], as it is able to transfer style onto buildings and cars realistically. Semantic style transfer [169] by obtaining a semantic map using a CNN is another interesting method. For the face, non-matching facial features need to self-align using customised semantic maps and the results are coarse compared to the real data. The example of neural style transfer is shown in Figure 3.9.

Face-centred style transfer is demonstrated by Shih et al. [170]. This model uses multiple

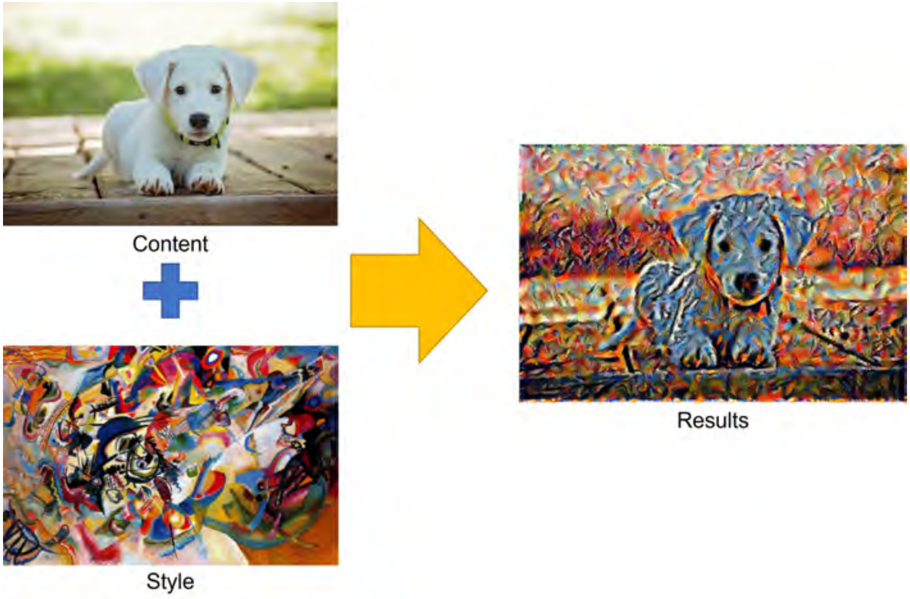


Figure 3.9: Neural style transfer. The resultant image has the structure of the content image and the style of the reference images. Images and method were adapted from Coursera.

energy maps which transfer only the colour of the style while retaining most of the facial features, which are well preserved. However, it is very sensitive to lighting, so if the source and reference do not have the same lighting, the results will become unrealistic. The generated faces remain visually recognisable from the original images despite the colour changes.

The current state-of-the-art human face style transfer is StarGAN. StarGAN [171] performs image-to-image translation for multiple domains using a scalable model. This model demonstrates its effectiveness in facial attribute transfer and facial expression synthesis. StarGANv2 [128] further enhances it by replacing domain labels with domain-specific style codes. Compared to the previous version [171], it has additional structures: a mapping network and a style encoder. The mapping network learns to transform random Gaussian noise into a style code, while the encoder extracts a style code from the reference image. StarGANv2 modifies the style of the source frame containing ME using a reference frame. This is performed by applying a style reconstruction loss (shown in equation 3.8) onto the generator to utilise the style code during the image generation process, referred to as reference-guided image synthesis.

$$L_{sty} = E_{x,y,z}[s - E_y(G(x, s))] \tag{3.8}$$

where L_{sty} is the style reconstruction loss, E is the encoder, G is the generator, x is the original image, y is the domain, z is the latent code and s is the style code.

3.4.12 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [172] are deep learning based generative models. The working principle of GANs is a zero-sum game by at least two neural networks (at least one generator and one discriminator) where the generator generates new data while the discriminator tries to classify real or fake samples. The generator tries to fool the discriminator while the discriminator learns to classify real and fake samples from the output of the generator and real samples. In an ideal scenario, the discriminator will have a classifying accuracy of 0.5. This means that the discriminator is getting correct predictions at random and the generator has succeeded in creating samples very similar to the real samples. This method works not only on images but on texts, speech and even videos.

Pix2Pix [173] is a conditional GAN that performs image-to-image translation. This method uses a generator which is an autoencoder with "U-Net" architecture and a discriminator (named PatchGAN) that penalises at the scale of patches. The PatchGAN makes the representation smaller and hence faster to run especially on larger images. Pix2pix is shown to work on a wide variety of things depending on the task and data assigned.

CycleGAN [174] is also an image-to-image translation generative method. This method can be thought of as an extended version of Pix2Pix that uses two generators and two discriminators simultaneously. The main novelty of this GAN is it learns to generate an image from one domain to another and back again to the original domain. This is performed using Cycle Consistency Loss.

StyleGAN [40] is a type of GAN which is capable to select and scale a specific control of the generation process. This is done by changes to the generator by including a novel mapping network which maps the points in latent space to an intermediate latent space. The intermediate latent space is then used to control the style of the generation.

CapsuleGAN [175] shows that CapsuleNet-based GAN exhibits better generative adversarial metrics on MNIST dataset and a lower error rate on semi-supervised classification tasks on both MNIST and CIFAR-10 datasets compared to the convolutional counterpart. The experiment is evaluated by comparing images generated by CapsuleGAN and its CNN equivalent.

Aesthetically, images generated by CapsuleGAN look cleaner and crisper. By using generative adversarial metric (a comparison metric between GAN models by pitting each generator against the opponent's discriminator), CapsuleGAN outperforms convolutional GAN on MNIST dataset while performing similarly on CIFAR-10. In semi-supervised classification by using Label Spread algorithm [176], the error rate of CapsuleGAN is lower in both MNIST and CIFAR-10. This method only implements CapsuleNet on the discriminator part of the GAN. However, the number of parameters of the CapsuleNet used is low which may indicate that CapsuleNet might not be well suited as the discriminator or modifications are needed for it to fully utilise its potential. Margin loss is used as it is the staple loss used in CapsuleNet.

Otberdout et al. [177] demonstrate a method that generates video clips of six basic facial expressions using a neutral face. It involves a conditional version of manifold-valued Wasserstein generative adversarial network (WGAN) that conducts motion generation on the hypersphere. The architecture consists of two GANs. MotionGAN is used to synthesize facial expressions from noises. The resulting motion encoded by the Square-Root Velocity Function is applied to the neutral face landmark which generates landmark sequences. TextureGAN subsequently transforms the landmark sequences to frame sequences with identity preserved. It is able to change the intensity of facial expressions by using a factor, I.

EmotionGAN [178] use the inverse of the generator to establish the mapping between the input and feature vector. The pre-trained generator can be used to synthesize a variety of expression images, the expression synthesis process is controllable. Pre-trained ResNet is used to extract features from an input image. These features are used to reconstruct face image using a generator. The resolution of the generated image is high (1024×1024 pixels). Classification accuracy across all emotion classes is 0.897. The author claimed a smooth generation of facial expression images but only a single image was shown.

Ling et. al. [179] perform facial expression editing using U-Net-based architecture, using relative AU to ensure expression is generated continuously. All generated facial expression intensity and AU are controllable. Instead of passing the features directly using the skip connections of U-Net, this method concatenates the relative AU extracted with the latent representation of different spatial resolution using a separate module. The author describes this module combines the features using a mechanism inspired by depth-wise convolution.

Attention-based facial expression generation is also attempted. Wang et al. [180] use a multi-level attention mechanism in the generation process, a employ self-attention layer to the encoder for long-range dependency and use a discriminator feature-based loss function to suppress artifacts. This method is able to produce facial expressions with different intensities on the targeted AU. It is also capable of performing facial expression transfer.

Zhang et al. [181] use GAN inversion which involves inverting the source images to the latent spaces of trained GAN. It conducts gradient-based optimisation using l2 loss and lpips loss to obtain the latent vector of the source image. Attentive Expression Embedding (AEE) module to embed vectors into different facial regions. This module controls the weight of vectors by disentangle AU vectors using an attention matrix. With AEE, the network is able to edit expression in a larger magnitude.

3.5 Performance Metric

Training the model is only the first step. We need to know the quality of the model to further finetuning, troubleshooting and comparing with other methods. Performance metrics are quantitative measures to evaluate the model. It gives us information on how well the model performs. The notations used in this section are true positive (TP), false positive (FP), false negative (FN) and true negative (TN).

3.5.1 Confusion Matrix

A confusion matrix is a useful tool for describing the performance of a classification model. It is a visualisation of the classification performance by tabulating them into TP, FP, FN and TN. The simplest form is binary classification visualised using a two-by-two matrix. The true predictions lie on the main diagonal of the matrix.

3.5.2 Accuracy, Precision, Recall and F1-score

Accuracy is defined as the number of accurate predictions over the total number of predictions. *Precision* is defined as the number of TP divided by the sum of the number TP and FP. It is the ratio of the number of correct positive predictions to the number of total predicted positives. *Recall*, also known as sensitivity or true positive rate, is defined as the number of TP divided by the sum of the number TP and FN. It is the ratio of the number of correct positive predictions to the number of total positive examples. *F1-score* is the harmonic mean of both precision and recall. All these performance metrics are shown in Equations 3.9, 3.10, 3.11 and 3.12.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.9}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.11}$$

$$F1\text{-}score = \frac{2TP}{2TP + FP + FN} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.12}$$

3.5.3 Interval Based Spotting Metric

ME spotting are done on either a frame-by-frame or interval basis. Currently, the widely accepted criteria is by looking at the overlap between the signal interval and ground truth using Intersect over Union (IoU), defined as equation 3.13. This method is first introduced in Micro-expression Grand Challenge (MEGC) 2019 [8]. It is used in MEGC2020 [182], MEGC2021 [121] and MEGC2022 as the standardised criteria for ME spotting.

$$\frac{Predicted \cap GT}{Predicted \cup GT} \geq k \tag{3.13}$$

where k is the minimum overlapping to be classified as true positive, GT represents the ground truth expression interval (onset-offset), and $Predicted$ represents the detected expression interval.

3.5.4 ROC Analysis

The Receiver Operating Characteristic (ROC) curve shows the trade-off between true positive rate and false positive rate. It is useful in skewed class distribution and unequal classification error evaluation. For simplicity, classification problems with only two classes will have four possible outcomes which are TP, FP, FN and TN. This can be conveniently represented in a two-by-two confusion matrix. The true positive rate is calculated by the number of positives correctly classified samples divided by the total positives. The false positive rate is calculated by the number of negative incorrectly classified samples divided by the total negatives. Ideally, a point nearer to the top-left corner of the ROC curve (TP rate higher, FP rate lower) is better as the prediction of the model predicted higher TP and lower FP simultaneously. For the case of points that have the same distance to the top-left corner, the points that are closer to the left side are more "conservative" than points closer to the right side as it only makes positive classification only with strong evidence with fewer false positives. The diagonal line of y=x represents the performance of a random classifier. If the classifier appears in the lower right (below the diagonal line), it performs worse than random guessing and vice versa. The area under ROC curve (AUC) is used as a method to compare the performance of ROC. It reduces ROC performance into a single scalar value which represents the expected performance. The higher the AUC, the greater the average performance. An example of ROC curve is shown in Figure 3.10.

ROC curves are insensitive to changes in class distribution. This is especially useful in realistic situations. Conventionally, we think large class skews are rare and unrealistic; in the real world, class skew is very common. Class skew with a magnitude up to 100 happens frequently.

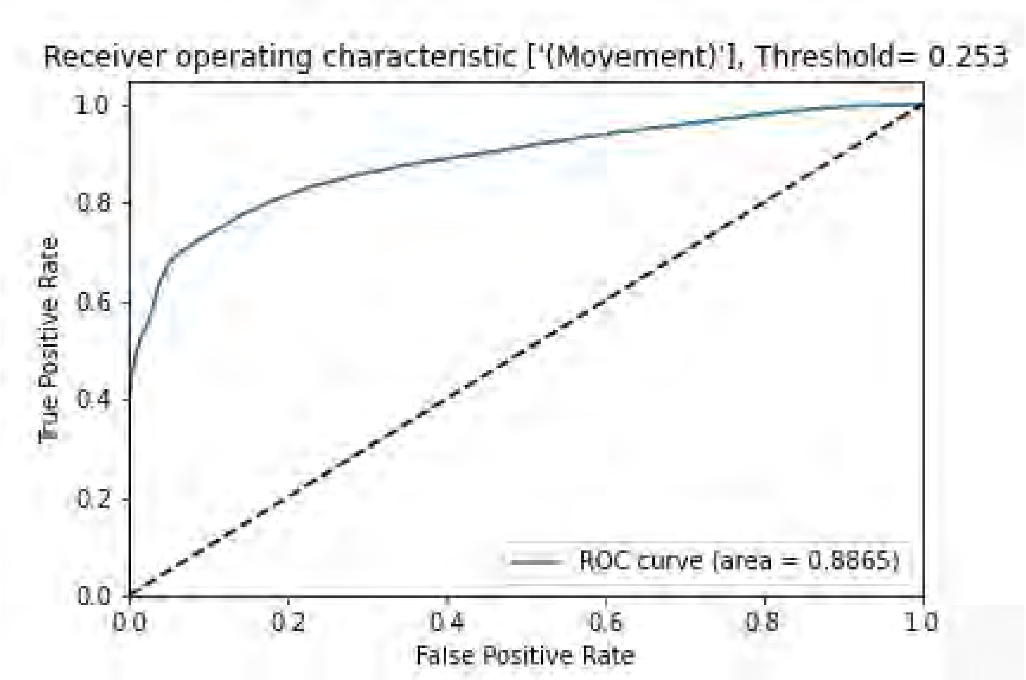


Figure 3.10: ROC curve example. The dotted line represents the performance of a random classifier. This roc curve is above this line which indicates this method performs better than random guessing. This figure also appears in Chapter 5.

3.5.5 Correlation Analysis

Correlation analyses are quantifiable methods that describe the relationship between two or more variables. In this thesis, we quantify the motion transferred onto a new stylised face using these metrics

Pearson's Correlation Coefficient shows the linear relationship between two sets of data. The range of this correlation coefficient is between -1 to +1. The formula of this correlation involves dividing the covariance by the product of standard deviation as shown in Equation 3.14.

$$r = \frac{\Sigma(x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\Sigma(x_i - \hat{x})^2 \Sigma(y_i - \hat{y})^2}} \tag{3.14}$$

where r is the correlation coefficient, x_i is the value of the x-variable, $\hat{x}$ is the mean of the x-variable, y_i is the value of the y-variable, $\hat{x}$ is the mean of the y-variable

Spearman's Correlation Coefficient [183] measures the monotonicity of two ranked variables. The variables in a monotonic relationship move in the same relative direction, but not necessarily at a constant rate. While the variables in a linear relationship move in the same direction at a constant rate. Similar to Pearson's correlation, the range of this correlation coefficient is between -1 to +1 too. The formula of this correlation coefficient is shown in Equation 3.15.

$$r = 1 - \frac{6\Sigma d_i^2}{n((n)^2 - 1)} \tag{3.15}$$

where r is the correlation coefficient, d_i is the difference between the two ranks of each observation, and n is the number of observations.

Due to the non-additive nature of correlation coefficients, we cannot compute the mean of the correlation coefficients by averaging them. To obtain the mean, each correlation coefficient must be transformed using Fisher's z-transformation [184] before averaging. The z-transformation is conducted using Equation 3.16.

$$z = \frac{1}{2} ln(\frac{1+r}{1-r}) = tanh^{-1}(r) \tag{3.16}$$

where z is Fisher's Z-score and r is the correlation coefficient.

3.5.6 Signal Processing

Signal processing is a method that removes unwanted noise from the raw output. This reduces the effect of unrelated information (noise) in influencing the predicted outcomes.

Savitzky-Golay filter operates by using a local least-squares polynomial approximation for signal smoothing. It involves sampling a fitted polynomial (which is identical to a fixed linear combination of the local set of input) [185]. This is conducted using a window that shifts by one step within the signal after each calculation. The scanning window has a size of $2m + 1$ (where m is any positive integer). The condition of this filter is where the polynomial of degree, n is less than $2m + 1$ [186].

Power Spectral Density (PSD) function is the Fourier Transform of the auto-correlation function of a signal. It shows the strength of energy as a function of frequency [187]. Generally, it shows which frequencies variations are strong and at frequencies variations are weak. It is useful in determining the cutoff frequency of a low pass filter by plotting the strength of each signal in terms of frequency and removing those that have frequencies higher than the signal.

Butterworth filter [188] can be used as a low pass filter. A low pass filter allows signals with a frequency lower than the cutoff frequency to pass through while attenuating signals with frequencies higher than the cutoff. An example of low pass filter is shown in Figure 3.11. The gain of the Butterworth filter is simplified in Equation 3.17, where $G(\omega)$ is the gain, ω is the angular frequency and n is the order of the Butterworth filter. The order of filter changes the behaviour of the transition band as shown in Figure 3.12. It can be tuned for cutting off signals using a sharper cutoff with the expense of greater gain changes over frequency within the transition band. The main advantage of this filter is it has a flat magnitude filter whereby signals with frequencies lower than the cutoff frequency do not undergo attenuation.

$$G(\omega) = \frac{1}{1 + \sqrt{\omega^{2n}}} \tag{3.17}$$

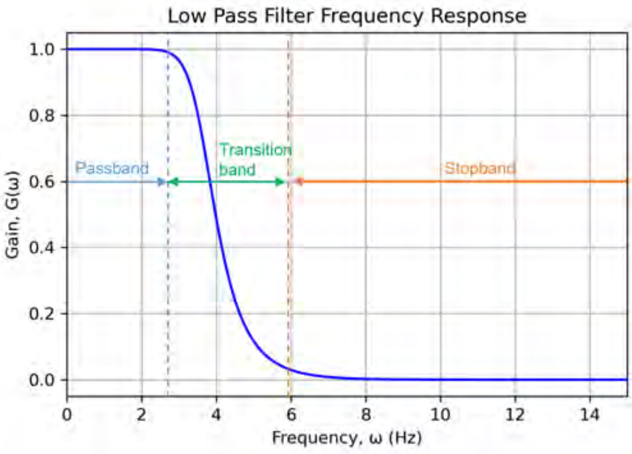


Figure 3.11: Example of Low Pass Filter Frequency Response. Signals with frequency within the passband will remain unchanged while signals with frequency within the passband will be completely blocked. Transition band as the name implied is the transition between passband and stopband.

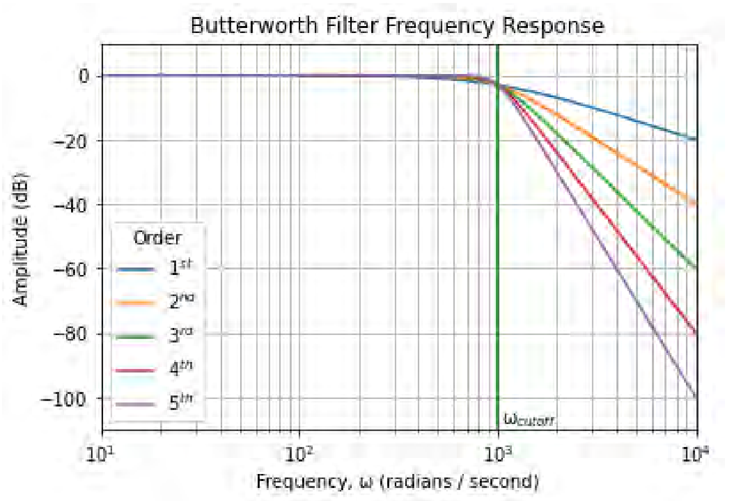


Figure 3.12: Frequency Response of Butterworth Filter with Different Order. Signals with a frequency higher than ω_{cutoff} are attenuated. The transition band shown is dependent on the order of the filter.

3.5.7 Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping (Grad-CAM) is a tool to analyse deep learning models. It visualises activation of CNN using the class discriminative localisation technique [189]. By using the gradient information flowing into the last convolutional layer, the importance of each neuron could be identified. It sets the gradient to "1" for targeted classes and "0" for the other classes.

The implementation of Grad-CAM is modified from Class Activation Mapping (CAM) [190]. CAM predicts the results of global average pooling, $\sum\limits_{x,y} f_k(x, y)$, performed on the activation units, $f_k(x, y)$ where k is the number of activation units of the last convolutional layer. The class score calculated by CAM is shown in Equation 3.18.

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_{x,y} \sum_k w_k^c f_k(x, y) \tag{3.18}$$

where w_k^c is the class feature weight and $f_k(x, y)$ is the activation unit

Grad-CAM uses a global-averaged-pooled gradient to obtain neuron importance weight α_k^c as class feature weight. This made Grad-CAM a more generalised version of CAM. The results of the generalisation enable complex cascade convolutional layers to be expressed in a visual map.

3.5.8 Image Quality Analysis

Image quality analysis is a measurement of the perception quality of an image. It can be used as a performance metric to determine the realism of the generated images by comparing the similarity with the original images. This analysis is used in our generative methods (in Chapter 6).

Inception Score uses Inception v3 model [191] trained on ImageNet to obtain a conditional probability of each image. This is done by computing the KL-divergence of the conditional and marginal probability distributions of the generated images. This metric measures the similarity of images. For GAN evaluation, this metric fundamentally disagrees with the subjective evaluation of human observers. The weakness of this method is it requires the compared images to be square and have a dimension of approximately 300×300 pixels. The accuracy of this metric is dependent on the trained model and distribution of the training images.

Frechet Inception Distance (FID) is a measurement of feature vector distance between real and generated images. It is often considered as a better version of Inception Score. Similar to Inception Score, FID score also uses Inception v3 model, the difference is it calculates the Frechet distance between real and generated images as multivariate Gaussian, instead of only based on the generated images as in IS. The lower the FID the better the image quality.

Peak signal-to-noise ratio (PSNR) is the ratio of the maximum possible power of a signal to the power of corrupting noise that affects the quality of the signal. This metric is commonly represented logarithmic-ally in decibel. It is calculated using the equation below:

$$\mathrm{PSNR} = 10 \cdot log_{10}(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}) \tag{3.19}$$

where MAX is the maximum possible pixel value and MSE is the mean square error of the difference between the original and generated images.

Structural Similarity Index (SSIM) [192] is a measure of the similarity between two images. This metric is a perception-based model which compares three key features (luminance, contrast and structure) between two images. This metric is computed using a local square window that processes the image pixel-by-pixel across the whole image. It is a full reference metric. It has a range between 0 to 1.

The operation of SSIM is described in Equations 3.20.

Luminance (of vector x):

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$$

Contrast (of vector x):

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{\frac{1}{2}}$$

Structure (of vector x):

$$s(x) = \frac{x - \mu_x}{\sigma_x}$$

Luminance comparison:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

Contrast comparison:

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

Structure comparison:

$$c(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

SSIM index:

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{3.20}$$

where μ is the luminance, σ is the standard deviation of the estimated signal contrast, and C is a constant to ensure stability. There are parameters (α, β, γ) that adjust the relative importance of the three features (luminance, contrast, and structure respectively) with conditions that $\alpha > 0$, $\beta > 0$ and $\gamma > 0$.

Natural Image Quality Evaluator (NIQE) [193] applies 36 identical natural scene statistic (NSS) features from P × P size patches by fitting them with the MVG model. The processed patches are then compared with the natural multivariate Gaussian (MVG) model. The "natural" in this section means models or images trained/taken from nature (e.g. trees, wild animals). A lower score indicates a better perceptual quality of an image. **Natural scene statistic (NSS)** features are low-order statistics of natural images. NIQE only uses the NSS features of natural images. **Multivariate Gaussian (MVG) Model** are computed from natural image patches fitted using an MVG density. This model uses standard maximum likelihood estimation to calculate the mean and covariance matrix of the model. This model was trained using copyright-free Flickr data and Berkeley image segmentation database.

The quality of the processed image is calculated as the distance between the quality-aware NSS feature model and the MVG using the following equation:

$$D(\nu_1, \nu_2, \Sigma_1, \Sigma_2) = \sqrt{(\nu_1 - \nu_2)^T (\frac{\Sigma_1 + \Sigma_2}{2})^{-1} (\nu_1 - \nu_2)} \tag{3.21}$$

where ν_1, ν_2 and Σ_1, Σ_2 are the mean vectors and covariance matrices of the natural MVG model and the processed images.

3.6 Summary

This chapter gives an in-depth review of the theories and techniques used in the following chapters. Chapter 4 consists of long video datasets that contains MEs and MaEs. Evaluation methods of the datasets are discussed. Chapter 5 is a machine learning method that detects ME and MaE using a novel framework that leverages the difference between these expressions. This method acts as the baseline for MEGC2021 and MEGC2022. Chapter 6 involves generative models. The first method is an advanced image augmentation using neural style transfer on an existing dataset (SAMM-LV). The second method is a facial expression sequence generation that is capable of generating spontaneous facial expressions without external guidance using two input sequences. Chapter 7 is an evaluation method on the generated sequences. It describes facial expression recognition methods using two objective tasks.

Chapter 4

New Long Video Datasets

4.1 Introduction

In Section 2.3, we discuss that ME dataset in the early stages are consisted of short clips containing posed expression. This does not represents the real life as ME is rare and happens more frequent in high stakes situations. The contribution of this chapter is to create more data that resembles real situation. A new dataset is curated based on the raw footage of SAMM [37], which contains both ME and MaE in long videos format. This dataset is named SAMM Long Videos (SAMM-LV).

Prior to this dataset, there is only one publicly available ME long video dataset, CAS(ME)^2. As deep learning requires a lot of sample to train, more useful, diverse data must be used for ME spotting system to work. Additional data is needed so that the trained algorithm can generalise better with new unseen data.

We further our effort in increasing the number of data by performing neural style transfer on SAMM-LV. The original identity of the participants was blended with the image style of the reference image. This increases the reproducibility of the facial movement of the source data by changing the identity while preserving the identity of the participants. Using this approach, a new synthetic dataset was produced, named SAMM-SYNTH.

There is an increasing need for long video dataset in ME research, where ME is a rare facial movement. It can co-occurs and even overlaps with MaE. Spotting task performed on a short

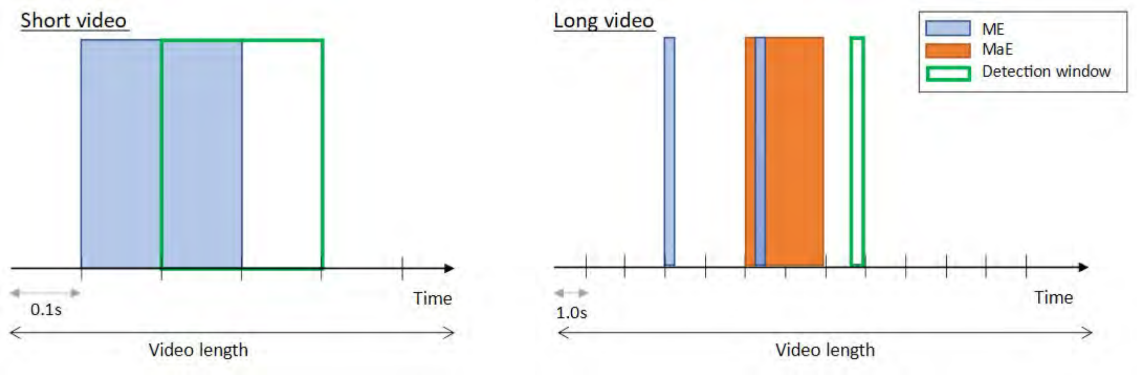


Figure 4.1: Visualisation of detection window on video with different length. The ground truth and detection window remain the same for both chart. For short video, the window of detection window has a higher chance to spot ME while for long video, the chance reduced significantly. Longer video increases the difficulty of spotting task. ME and MaE can also overlap which makes the task more realistic and challenging.

clip that contains ME is easier and unrealistic as has a higher chance of detection. With a longer video, the task becomes more challenging as there are more neutral frames which reduces the chance of detection. This is demonstrated in Figure 4.1.

4.2 SAMM Long Videos

To facilitate Facial Micro-expression Grand Challenge 2020 (MEGC2020) [182] in the ME and MaE recognition and spotting task, SAMM-LV is introduced. This dataset consists of 147 long videos with 343 macro-expressions and 159 MEs. It was captured in a frame rate of 200 fps. Each video is FACS-coded with detailed AUs. The onset, apex and offset frames for every expressions are also annotated. Two examples of facial expressions is shown in Figure 4.2. The top row illustrates a micro-expression of brief AU12 with low intensity and the bottom row shows a macro-expression of AU12 with high intensity.

The main advantages of this dataset are it is the highest resolution, highest frame rate and most ethnically diverse (participants with 13 different ethnicities) ME dataset till date. The high frame rate ensures that information on the expression was preserved while the diversity of the dataset improves generalisation of the algorithm.

4.2.1 Dataset Analysis

OpenFace [4] is used to analyse and provide a baseline result for this dataset. It is a software for facial analysis and was used mainly for face alignment and detection of AUs. Further details

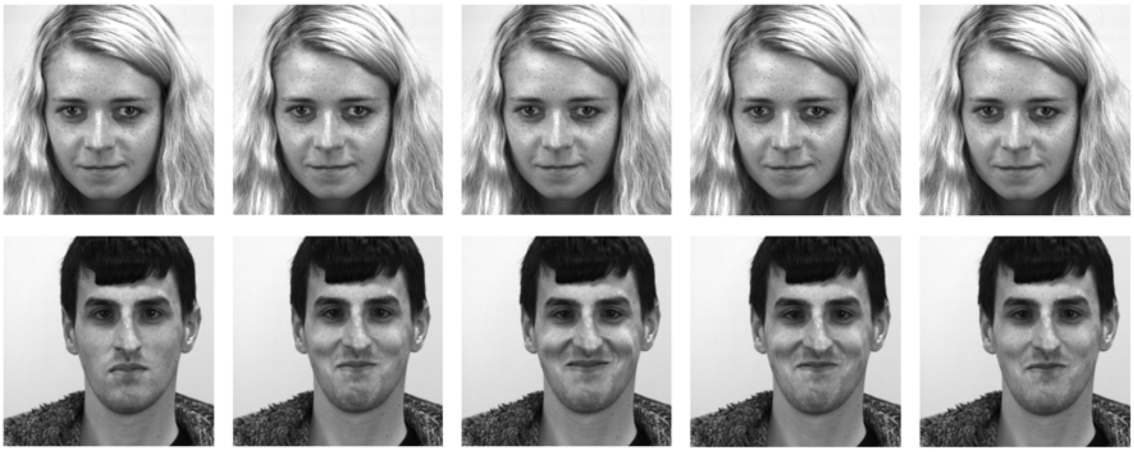


Figure 4.2: Two examples of facial expressions from SAMM Long Videos dataset: (Top) Micro-expression; and (Bottom) Macro-expression. Both expressions with AU12. On apex frame (middle), the macro-expression shows higher intensity and visibility when compared to the micro-expression.

of this toolkit can be found in Section 3.2.3. In this analysis, OpenFace was used to map the face texture by scaling and in-plane rotation. The output image has a dimension of 112 × 112 pixel with interpupilary distance of 45 pixels. The illustration of the original SAMM image, the facial landmarks and the aligned face image is shown in Figure 4.4. Other than face alignment, OpenFace has the ability to detect the presence and intensity of AUs. The presence of AU is encoded 0 as absent and 1 as present while the intensity of AU is represented on a 5-point scale. AU presence and AU intensity are treated as two different task. The former is a classification task trained using Support Vector Machines, the latter is a regression task trained using Support Vector Regression. We take advantage of these functions and provide a quantitative evaluation for our dataset.

Detection of facial movements was done by first combining the intensity of all the detected AUs and normalised it to a scale of 1. Next, Savitzky-Golay filter [186] is implemented for noise reduction. The onset and offset frame of the smoothed signal are obtained using a custom algorithm modified from [194]. It uses Daubechies wavelet [195] with scaling function of 2 and level 3 signal decomposition for signal smoothing. The onset and offset frames detected are compared with the ground truth. The algorithm also defines peak as a point that is higher than the 7 points that come immediately before and after it. Furthermore, a threshold of 75th percentile was used to filter out low local peaks. For this dataset, since the frame rate is 200 fps, a threshold of 100 frames (0.5 s) was selected to classify the spotted intervals into

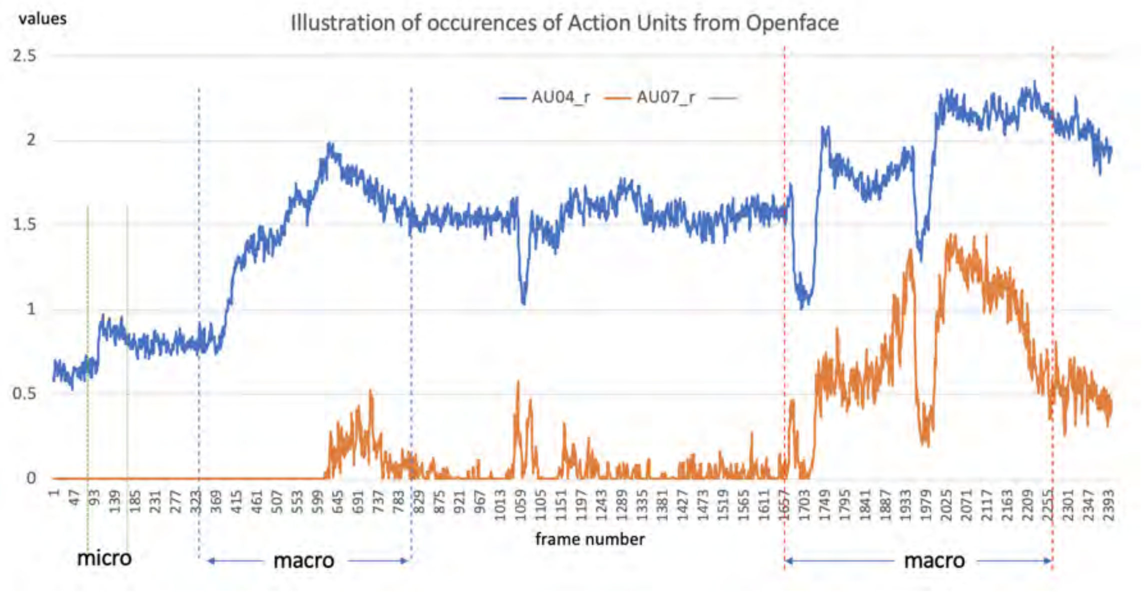


Figure 4.3: Two AUs on long video clips, the blue line is AU4, and the orange line is AU7. There are one micro-expression and two macro-expressions found on this video clip.

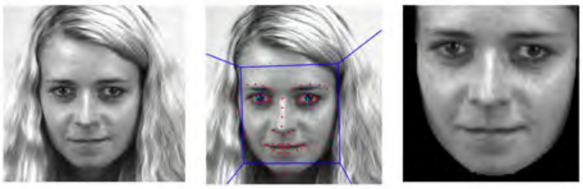


Figure 4.4: An illustration of preprocessing steps using OpenFace: (Left) Original SAMM image; (Middle) Facial Landmark Detection; and (Right) Cropped face as region of interest (ROI).

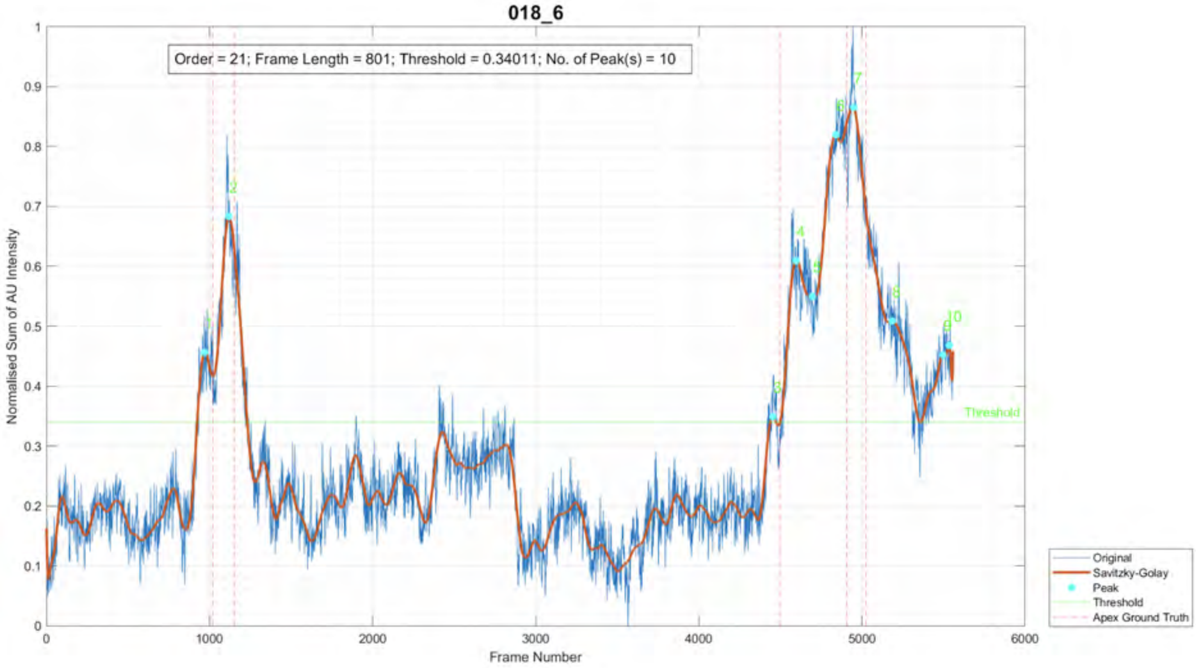


Figure 4.5: Plot of Subject 018_6 for the normalised sum of AU intensity (blue line) and the filtered signal (red line) with peaks annotated. A threshold value of 75th percentile of normalised AU intensity was used to filter out the peaks originated from noise.

macro-expressions (>0.5 s) or micro-expressions (≤0.5 s). The normalised sum of AU intensiy is illustrated in Figure 4.5. The overlapping of the prediction and ground truth is checked by the Equation 3.13 (a criteria for spotting challenge in [196]).

Move-to-neutral Ratio

There is a distinct difference between long video datasets which is the movements in each long video varies. To quantify these difference, a new metric named move-to-neutral ratio is introduced. As each subject of the dataset used has different numbers of frames with movement (ME or MaE) and duration of recorded videos, the proportion of movements to video duration of each subject is different. This can result in easier predictions on some subjects as they have more movements and vice versa. The move-to-neutral ratio of each subject is shown in Figure 4.6. The average move-to-neutral ratio for both datasets are approximately 0.40 (SAMM-LV) and 0.05 (CAS(ME)^2), which shows that both are imbalanced as most of the videos consist of neutral frames. However, this metric is solely based on the movements labelled in the ground truth, which consists of MaE and ME. Other movements such as head movements might not be included.

CAS(ME)^2 has an average move-to-neutral ratio about 10 times smaller than SAMM-LV.

This indicates that CAS(ME)^2 contains about 10 times more neutral frames compared to SAMM-LV making it a harder dataset to spot ME or MaE.

4.2.2 Benchmark Results

We report the results for both AU presence analysis and facial movement spotting.

For **AU presence analysis**, the *Accuracy* is shown in Table 4.1. To compare the performance of OpenFace on micro- and macro-movements detection, we analyse the micro-only videos and macro-only videos. The comparison between the performance metric of videos containing only macro- or micro-expression is shown in Table 4.2. All videos used in this comparison contains either with macro- or micro-expressions which gives a fairer evaluation in the detection accuracy of these two classes. There are 18 micro-only and 68 macro-only videos in this dataset. As we observe, micro-expression detection has a lower performance compared to macro-expression detection. The AU recognition algorithms (AU presence and AU intensity) of OpenFace are trained using facial expression datasets (DISFA [197], SEMAINE [198] and BP4D [199]) which makes OpenFace performs better in macro-expression compared to micro-expression. We did a separate studies by selecting macro-only and micro-only videos. OpenFace shows a significant performance drop in micro-only videos with *Accuracy* of 0.2903 in contrast to 0.4502 in macro-only videos.

For **facial movements spotting**, our results is compared with the baseline of MEGC2020 challenge are shown in Table 4.3. The overall results outperformed the baseline result in all three performance metrics of $Precision$, $Recall$, and $F1-Score$. This shows that our approach of summation of total AU intensity can be provide an estimation in spotting ME and MaE of the face.

In both results, macro-expression spotting yields the highest F1-score. This confirms that macro-expression spotting is easier compared to micro-expression spotting. It can also be explained as macro-expression has longer interval (> 0.5s) and hence easier to be spotted by the spotting algorithm. Moreover, OpenFace was trained on facial expressions datasets, which have no labelled micro-expressions in the training set, which explain the low F1-score in spotting micro-expressions.

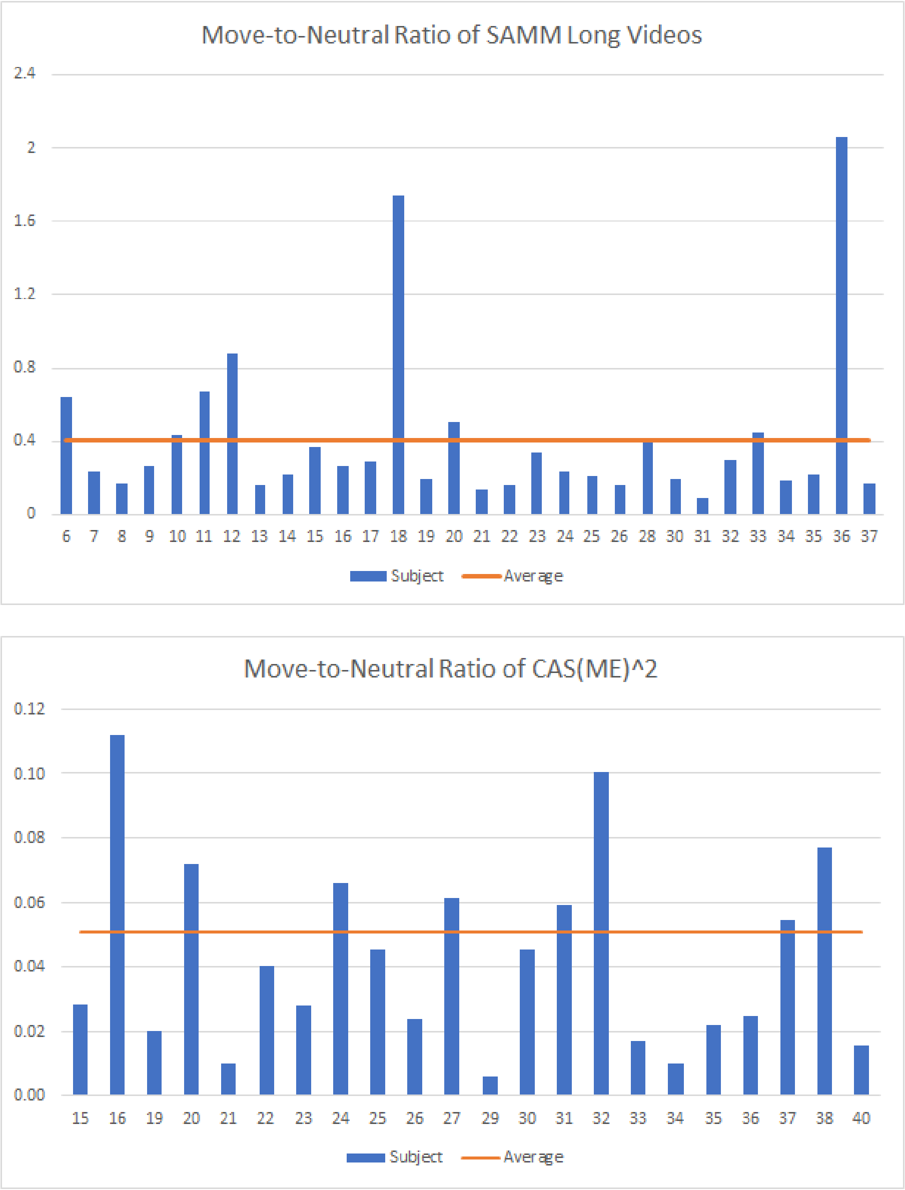


Figure 4.6: Move-to-Neutral Ratio for each subject of SAMM-LV and CAS(ME)^2. It shows that every subject has different relative number of movement (ME or MaE) to neutral frames. From the average move-to-neutral ratio, SAMM-LV is higher compared to CAS(ME)^2. *x-axis labels are the subject indices of the datasets.

Table 4.1: Accuracy of AU presence analysis in long videos using OpenFace

	Micro	Macro	Combined
Matched AU	80	245	278
Ground Truth	172	501	590
Accuracy	0.4651	**0.4890**	0.4712

Table 4.2: Comparison between AU presence analysis of videos containing only micro- or macro-expression

	Micro only videos	Macro only videos
Matched AU	9	104
Ground Truth	31	231
Accuracy	0.2903	**0.4502**

4.3 SAMM-SYNTH

Long video datasets of facial macro- and micro-expressions remains in strong demand with the current dominance of data-hungry deep learning methods. To date of this publication, there are only two related datasets with ME and MaE, which are SAMM Long Videos (SAMM-LV) [31] and CAS(ME)^2 [30]. Hence, there is a high motivation to obtain more data. The number of samples of MEs are also not sufficient enough for generation process. Moreover, there is a lack of performance metrics to quantify the generated data. A new approach of generating synthetic long videos with MEs using style transfer is introduced and assessment methods (both quantitative and qualitative) to measure the quality of the generated data. Generating new data using style transfer can be consider as an advanced image augmentation which modifies the appearance of the subjects without changing the facial movements. To achieve this, the state-of-the-art generative adversarial network style transfer method – StarGANv2 is used. Using StarGANv2 pre-trained on the CelebA dataset, the style of a reference image from SAMM long videos (a facial micro- and macro-expression long video dataset) was transferred onto a source

Table 4.3: Results of macro- and micro-spotting in SAMM Long Videos compared to the baseline of MEGC2020.

	Our Results			**Baseline of MEGC2020**[200]		
Expression	macro-expression	micro-expression	overall	macro-expression	micro-expression	overall
Total number	343	159	502	343	159	502
TP	**172**	6	**178**	22	**29**	51
FP	328	71	399	334	1407	1741
FN	171	153	324	321	130	451
Precision	**0.3440**	**0.0779**	**0.3085**	0.0618	0.0202	0.0285
Recall	**0.5015**	0.0377	**0.3546**	0.0641	**0.1824**	0.1016
F1-score	**0.4081**	**0.0508**	**0.3299**	0.0629	0.0364	0.0445

image of the FFHQ dataset to generate a synthetic dataset (SAMM-SYNTH).

This approach preserves facial motion and changes the identity of the participant to produce new data. It is fully automated and able to generate unlimited amount of data without any supervision. This method also has an added benefit of reusing the ground truth labels without the need of additional FACS coding.

4.3.1 Method

We take advantage of the ability of StarGANv2 [128] to extract style from one image and transfer onto a target image. The ME and MaE movements remains the same while the identity of the original image was changed. This is an advanced image augmentation approach, which essentially creates unique face features while preserving facial movements. The overall pipeline of our method (including analysis) is shown in Figure 4.7.

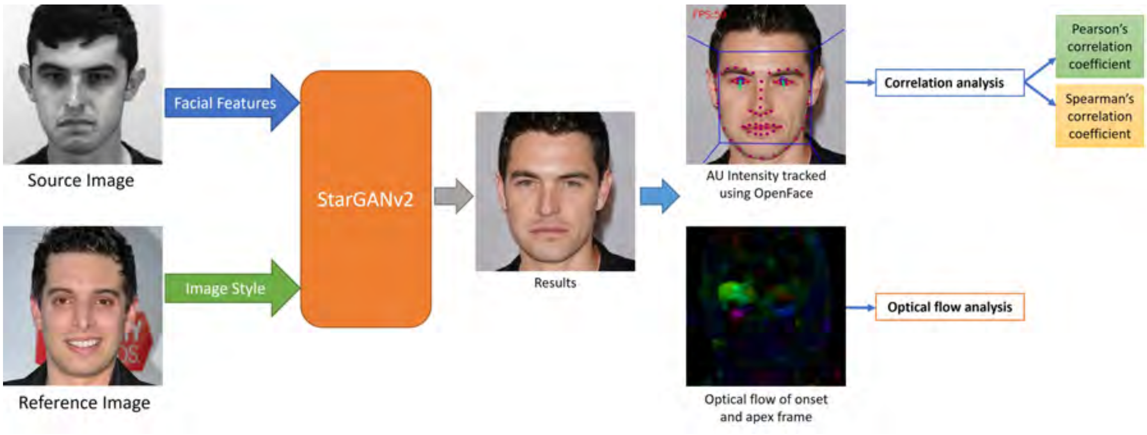

Source Image
Facial Features
StarGANv2
Reference Image
Image Style
Results
AU Intensity tracked using OpenFace
Correlation analysis
Pearson's correlation coefficient
Spearman's correlation coefficient
Optical flow of onset and apex frame
Optical flow analysis

Figure 4.7: Overall pipeline of our method with analysis. StarGANv2 is used to modify the "style" of participant from SAMM-LV dataset [31]. The synthetic image exhibit the style extracted from the reference image (from FFHQ dataset [40]) while maintaining the facial features of the source image. Facial Action Units (AUs) of the synthetic image is measured using OpenFace [4]. The AUs and optical flow of the synthetic image are compared with the original source image.

Datasets

Facial micro- and macro-expressions dataset, SAMM-LV (details in Section 2.3), is used as source frames, which is the target for style transfer. The generative algorithm (StarGANv2) uses CelebA dataset [201] as the training set. CelebA is a face-based dataset which contains ten thousand unique identities, where each of them has twenty images. The reference frames (containing 14 female and 14 male adults) are selected from FFHQ dataset [40], another face-based dataset, so that more variety of generated faces can be produced. This can also prevent StarGANv2 from using its training set for style transfer and demonstrate the robustness of style transfer of the model.

Network Architecture

StarGAN [171] is able to perform image-to-image translation for multiple domains using a scalable model. This model demonstrates its effectiveness on facial attribute transfer and facial expression synthesis. StarGANv2 [128] further enhances it by replacing domain labels with domain-specific style codes. Compared to the previous version [171], it has additional structures: a mapping network and a style encoder. The mapping network learns to transform random Gaussian noise into a style code, while the encoder extracts a style code from the reference image. StarGANv2 modifies the style of the source frame containing ME using a reference frame. This is performed by applying a style reconstruction loss (shown in equation 3.8) onto the generator to utilise the style code during the image generation process, referred to as reference-guided image synthesis.

In our experiment, we use StarGANv2 pre-trained on the CelebA dataset [201] for reference-guided image synthesis.

4.3.2 Results and Discussion

We generated a synthetic dataset, named SAMM-SYNTH, using reference-guided image synthesis of StarGANv2 on SAMM-LV dataset. To evaluate the quality of the generated dataset, first we perform quantitative analysis using facial action units (AUs) detected by OpenFace. We conduct a correlation analysis on the AUs in SAMM-LV and SAMM-SYNTH. To compare the visual appearances, we perform qualitative analysis that involves the use of optical flow.

The results of StarGANv2 reference-guided image synthesis can be seen in Figure 4.8. We observed that the facial attribute follows the source image (taken from SAMM-LV) while the style follows the reference image (taken from FFHQ dataset). We conducted style transfer on each participant of SAMM-LV. As a result, SAMM-SYNTH consists of 147 long videos with 15 female and 15 male participants. We generated most synthetic data by following their original identified gender (other than 1 female participant, due to excessive artifacts). We also found that inter-gender style transfer is possible. An interesting observation is that we found StarGANv2 tends to generalise female participants to have long hair. This may be caused by the training set (CelebA) which consisted of mostly female participants with long hair.

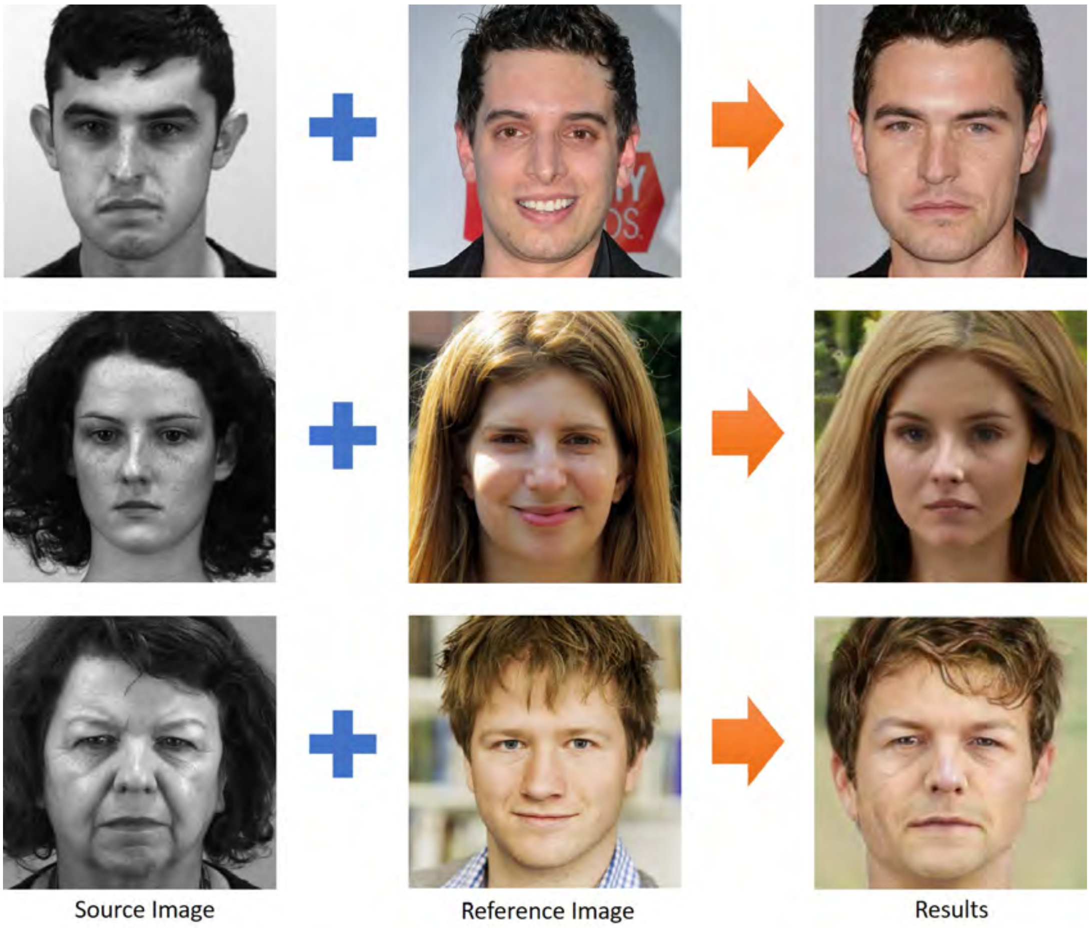


Figure 4.8: Style transfer using StarGANv2, also known as reference-guided image synthesis. The result exhibits facial features of the source image (taken from SAMM-LV dataset [31]), while taking the style extracted from the reference image (taken from FFHQ dataset [40]). StarGANv2 is capable of transferring style for both gender realistically. Inter-gender style transfer is also possible as shown in the bottom row.

Action Unit Analysis using OpenFace

OpenFace 2.0 [4] is a facial analysis toolkit that is capable of performing facial landmark detection, head pose estimation, facial AU recognition and eye-gaze estimation. OpenFace uses Convolutional Experts Constrained Local Model (CE-CLM) [132] for facial landmark tracking and detection. CE-CLM uses a deep convolutional neural network for the 84-point facial landmark detection. Based on the facial landmarks, the AU intensity of both the original and synthetic data are measured.

Two selected AU intensities measured by OpenFace for both original and synthetic videos are shown in Figure 4.9. They were smoothed using Savitzky-Golay filter [186]. The rise and falls of the AU intensity indicates movement on each particular AU. We observed that the AU12 movement is better replicated compared to AU4. Another reason may be resorted because AU12 (lip corner puller) has a bigger movement range when compared to AU4 (brow lowerer).

We analyse the similarities of facial movements between SAMM-LV (original) and SAMM-SYNTH (transferred) using Pearson's and Spearman's correlation [183]. To analyse the quality of AU transferred, each AU intensity (original and synthetic data) measured detected by OpenFace are transformed into Z-score using Fisher's Z-transformation [184] as in Equation (3.16). The average values of Z-score are calculated and converted to correlation coefficients (Pearson's and Spearman's coefficients). The reason for using Z-score when calculating the average is that correlation coefficients are non-additive. Hence, a sample-size weighing (e.g. Fisher's Z-transformation) must be applied before averaging.

We tabulate the results sorted by AU and by participant in Tables 4.4 and 4.5, respectively. Table 4.4 compares the Pearson's correlation coefficients sorted by AU, we observed that AU45, AU12 and AU6 are better replicated as they show higher Pearson's correlation coefficients of 0.92, 0.74 and 0.72, respectively. Based on OpenFace benchmark, when compared to the ground truth labelling, OpenFace performs better in AU4, AU12 and AU25 with a Pearson's coefficient of 0.70, 0.85 and 0.85 respectively, on a facial expression dataset (DISFA dataset [197]). By comparing Pearson's coefficients of our experiment and OpenFace benchmark, we have a high confidence that AU12 is the best transferred AU. This is proved whereby OpenFace has high accuracy and our experiment shows high correlation in AU12.

In Table 4.5, we observed that the overall mean Pearson's correlation is higher than Spearman's correlation. This implies that our data is more linear than monotonically correlated, as Pearson's correlation measures the linearity of the data correlations while Spearman's measures the monotony of the data correlations. Figure 4.10 further confirms our claim as the Spearman's distribution is skewed right, indicated a lower boundary in the coefficients.

We identified participants with glasses (participant 010, 020, 021, 032 and 037) and did a separate comparison by removing them from the whole dataset. In Figure 4.11, we can see that most of the Pearson's coefficient improves for the upper face AUs. The only AUs that do not improve are AU6 and AU7. These two AUs can affect the eye corners and muscles

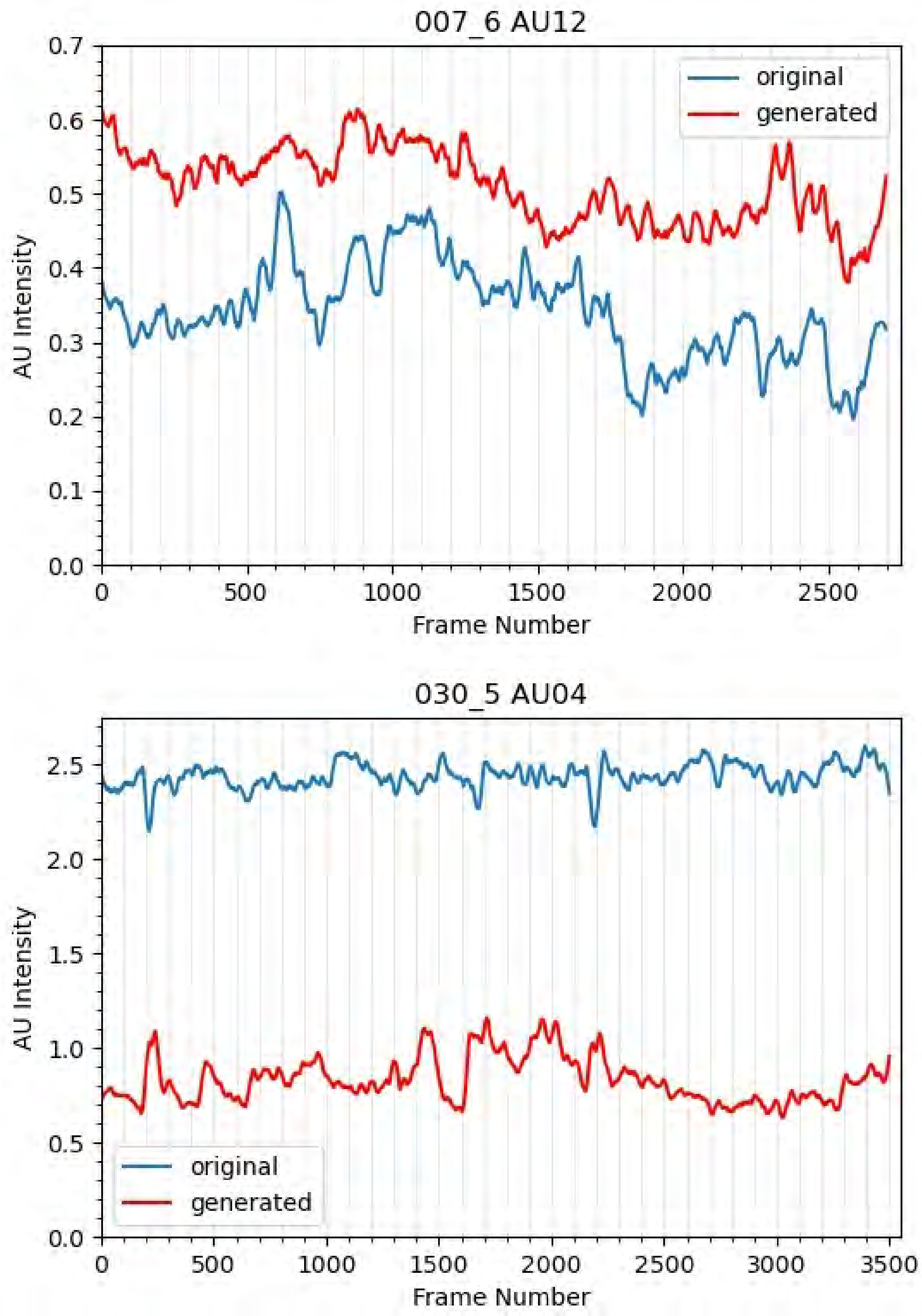


Figure 4.9: Illustration of AU intensities measured by OpenFace for selected videos of 007_6 and 030_5 of SAMM-LV and SAMM-SYNTH. AU12 is amplified, while AU4 is softened in the synthetic data. According to OpenFace's documentation, AU intensity has a range of 0 (not present) to 5 (maximum intensity). However, the AU intensity is standardised based on the initial frame. The changes in AU intensities indicate movements of that AU which is what we are interested in.

Table 4.4: Analysis of Action Units (AUs) detected by OpenFace on original SAMM-LV participants and SAMM-SYNTH synthetic videos. The bolded values are the top-3 highest Pearson's correlation coefficients. Pearson's correlation coefficients show strong positive correlation for AU6, AU12, and AU45, which are well preserved compared to the other AUs. "Benchmark" indicates the Pearson's coefficients of OpenFace detection versus ground truth labelling on DISFA dataset [197], which reported that OpenFace has better performance on AU4, AU12, and AU25 detection.

Action Unit	SAMM-LV vs SAMM-SYNTH	Benchmark
AU1	0.40	0.64
AU2	0.26	0.50
AU4	0.25	**0.70**
AU5	0.27	0.67
AU6	**0.72**	0.59
AU7	0.33	-
AU9	0.38	0.54
AU10	0.28	-
AU12	**0.74**	**0.85**
AU14	0.42	-
AU15	0.26	0.39
AU17	0.15	0.49
AU20	0.28	0.22
AU23	0.40	-
AU25	0.26	**0.85**
AU26	0.20	0.67
AU45	**0.92**	-

around the eye socket. The outer frame of the glasses might exaggerate the movement in the generation process resulting in higher correlation. This presents a challenge for style transfer on participants with glasses.

Optical Flow Analysis

We use optical flow to visualise the facial movements and their intensities in both original and synthetic data. Farneback optical flow [144] is used to calculate the frame differences between the onset and apex frame of ME related movements as shown in Figure 4.12. In image sequences of facial expression, the onset frame is the frame where the expression starts and apex frame is where the expression is at the highest intensity. For the optical flow method, we assume that all participants were captured under uniform lighting at all times. The direction

Table 4.5: Analysis of the quality for each participant of SAMM-LV versus SAMM-SYNTH. Both Pearson's and Spearman's correlation are compared. The degree of freedom, *df*, for the correlation coefficients is 14.

Subject	**Pearson**		**Spearman**	
	r	***p-value***	***r***	***p-value***
006	0.58	0.006	0.44	0.016
007	0.38	0.023	0.32	0.040
008	0.33	0.018	0.11	0.038
009	0.22	0.039	0.13	0.057
010	0.38	0.045	0.27	0.026
011	0.57	0.013	0.44	0.016
012	0.59	0.021	0.38	0.005
013	0.25	0.028	0.17	0.050
014	0.49	0.009	0.32	0.016
015	0.60	0.007	0.41	0.034
016	0.19	0.040	0.14	0.061
017	0.13	0.041	0.13	0.039
018	0.51	0.023	0.40	0.008
019	0.40	0.022	0.25	0.034
020	0.61	0.032	0.47	0.015
021	0.20	0.035	0.15	0.039
022	0.41	0.016	0.28	0.021
023	0.44	0.005	0.37	0.044
024	0.04	0.026	0.05	0.048
025	0.51	0.037	0.39	0.022
026	0.55	0.027	0.27	0.025
028	0.56	0.021	0.27	0.022
030	0.27	0.056	0.14	0.030
031	0.16	0.068	0.17	0.080
032	0.22	0.051	0.15	0.076
033	0.51	0.029	0.35	0.038
034	0.36	0.052	0.21	0.068
035	0.33	0.038	0.25	0.021
036	0.39	0.012	0.31	0.028
037	0.19	0.041	0.16	0.071
Mean	0.39	0.029	0.27	0.036
Std. Dev.	0.19	0.016	0.12	0.020

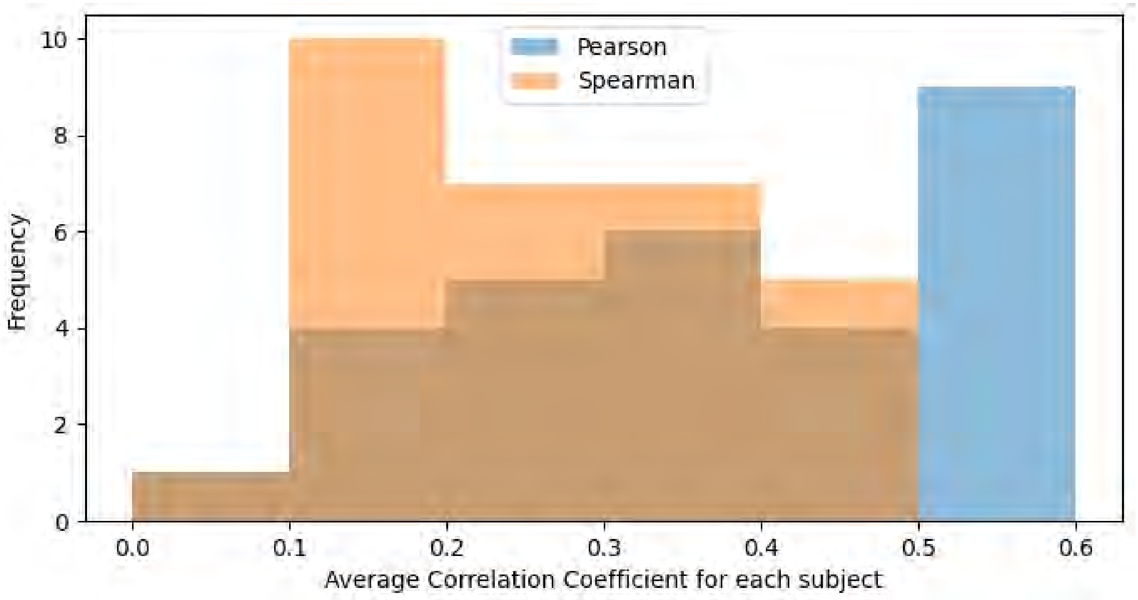


Figure 4.10: Histogram of correlation coefficients on all participants of SAMM-LV and SAMM-SYNTH. Both histograms use 6 bins. We observed that Spearman's coefficients are more right-skewed relative to Pearson's coefficient. This implies that our results are more linear than monotonic correlated.

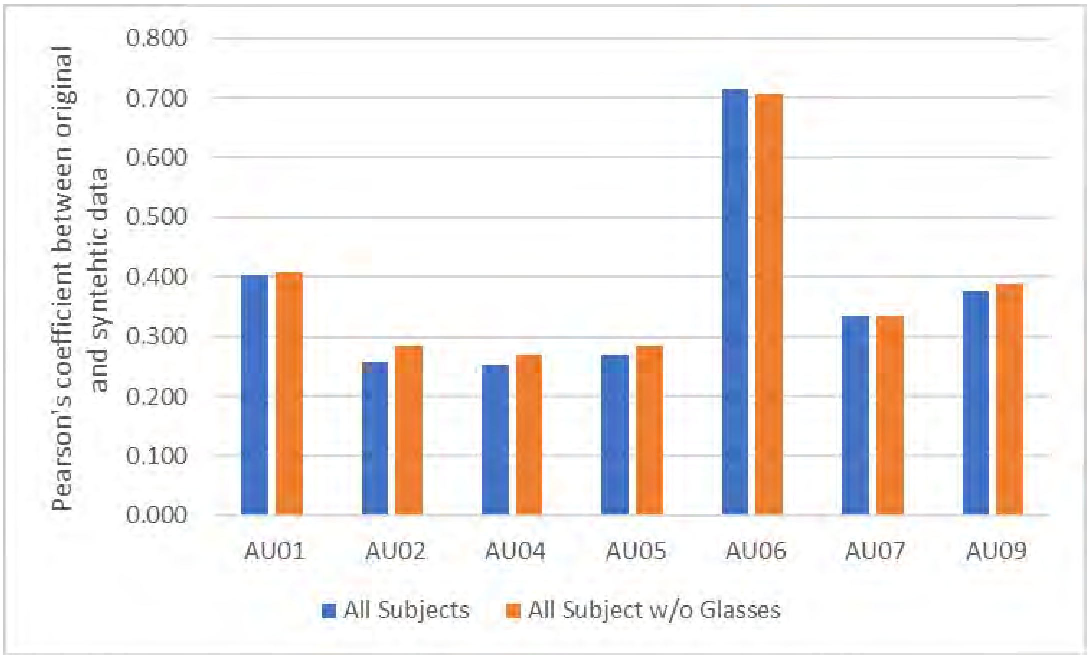


Figure 4.11: Comparison of Pearson's correlation of upper face AUs between all participants and those without glasses. By excluding participants with glasses, most of the upper face AUs improved.

and intensity of the movement is determined by the hue and brightness of the HSV colour model. For comparison, the original images were scaled down to 256×256 pixels to match the synthetic images. In AU12 movement, we observed that the right lip corner movements in both original and synthetic pairs are well replicated (bright spot on the optical flow images). In AU4 movement, the eye brow movements are less distinct, while the eye movements are better replicated. In both optical flow of synthetic data, we noticed that additional frame differences are observed around the jaw region, although it is not visible to the human eye, it was captured in the optical flow analysis.

4.3.3 Advantages

The main advantage of our approach is that the generated faces have a new style. This allows a neural network to be trained or validated on unseen data, which can potentially result in better model generalisation. It has an added benefit of protecting the identity of the original participant while generating new data. GAN generated methods [110], [120] require substantial labelled data. For our case, due to the limited MEs and MaEs long video datasets, generation could be a challenging task. Style transfer resolves this issue as it primarily transfers the facial movements without the need of training new model or data annotation. Moreover, style transfer is a simple and convenient method of data augmentation. By altering the reference frame alone, we are able to create new faces with ME or MaE included in the synthetic video. We can also generate cross-ethnic faces (as participant 030 in Figure 4.12) to increase the diversity of ME dataset, which primarily consists of participants from one particular country and ethnic background [202]. It is known that demographic imbalance of the training dataset will cause external biases on the trained models [203]–[205] that results in inaccurate and erroneous predictions. With the ability to generate wide range of participants with different demographics, this can alleviate the issue. Potentially, this will help to improve the generalisation of the deep learning models. It is also relatively low in computational cost. By training StarGANv2 once, we are able to generate an unlimited amount of style transfer data with very little computational cost.

4.3.4 Limitations and Challenges

We conduct extensive tests on our synthetic data by changing the reference images for each participant of SAMM-LV. We found that the background of the reference image is one of the

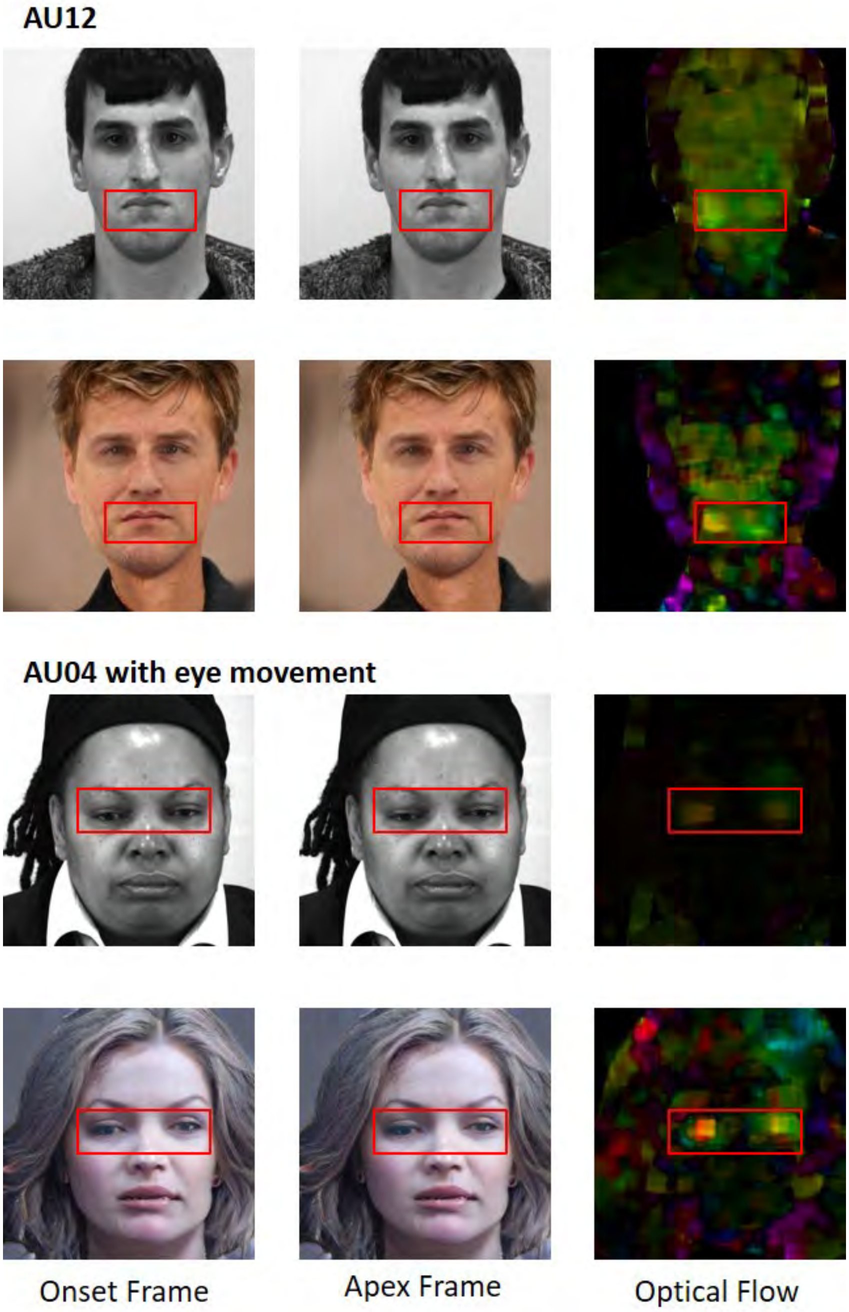


Figure 4.12: Optical flow comparison of onset and apex frames between original and synthetic MEs. Each respective AUs are shown in the red bounding boxes.(**Top**) For movement 007_6_5, AU12 is involved. (**Bottom**) For movement 030_5_1, AU4 and slight eye movement are involved. The hue represents direction, while the brightness represents intensity of the movements. Note: Original images were scaled to 256×256 pixels for fairer evaluation.

source of artifacts. Non-uniform background creates patches of artifacts which are not realistic. Hence, all the reference images are selected with a uniform background. However, there are still blob-like artifacts in some cases (source might originate from the generator that learnt unnatural distributions that can fool the discriminator). Next, source images with facial accessories (e.g. glasses) create unrealistic images. StarGAN treats glasses as a facial feature and attempts to blend it onto the face. Not only are the images are not suitable for real life applications, our separate analysis, which only includes participants without glasses (in Figure 4.11), shows improvement in the similarity of AUs transferred. This shows that the facial movement of participants with glasses are not well transferred. Facial accessories come in various sizes and shapes, which is challenging to transfer. They are also not well-represented in StarGAN's training set. The artifacts mentioned are shown in Figure 4.13. There are 5 participants with glasses and 1 participant with one video with glasses. Out of 147 videos, 25 videos contains artifacts caused by transferring facial movements on participants that wore glasses. A potential solution is to retrain the model on a new training dataset with facial accessories or inpaint the glasses before the style transfer process.

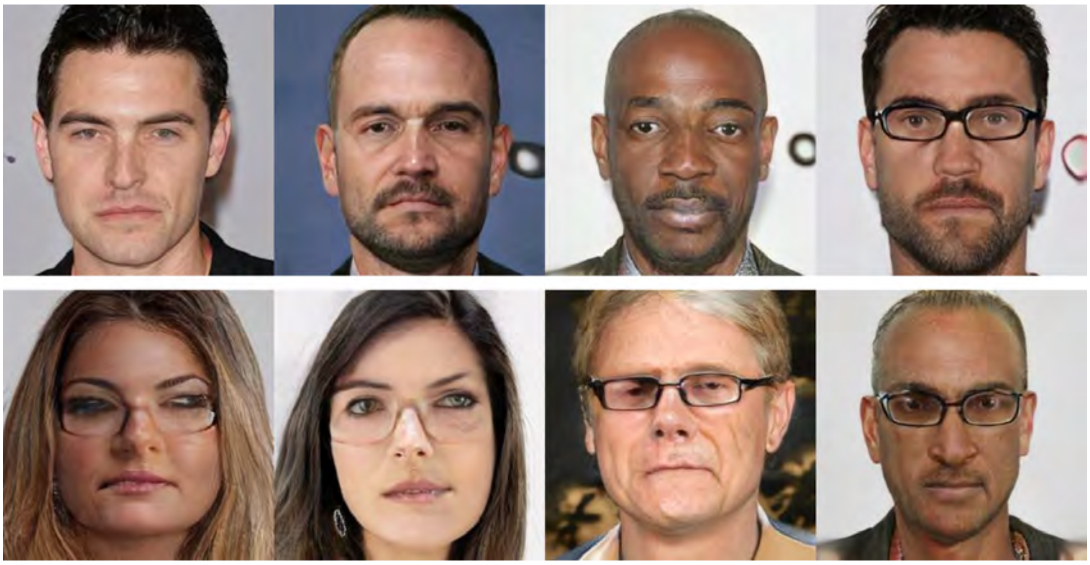


Figure 4.13: Artifacts present in synthetic images. (**Top**) Background artifacts of blob-like structure. (**Bottom**) Participants with glasses are not realistically generated.

We noticed that when eye blinks occur in the source images, the hair structure of the synthetic images was slightly altered. This might be caused by the pre-trained weights. The majority of the face images in CelebA are faces with open eyes, hence, it is reasonable to assume that when a participant blinks, the model wasn't able to preserve the face structure well, especially the hair. Selected examples with this issue are shown in Figure 4.14.

All the AUs of original and synthetic data are positively correlated. However, different AUs have different range and sensitivity, which may explain the low correlation coefficients in some AUs. The source images from SAMM-LV do not have a balanced distribution across all AUs, which may result in an uneven comparison that skew the results towards AUs that are more common.

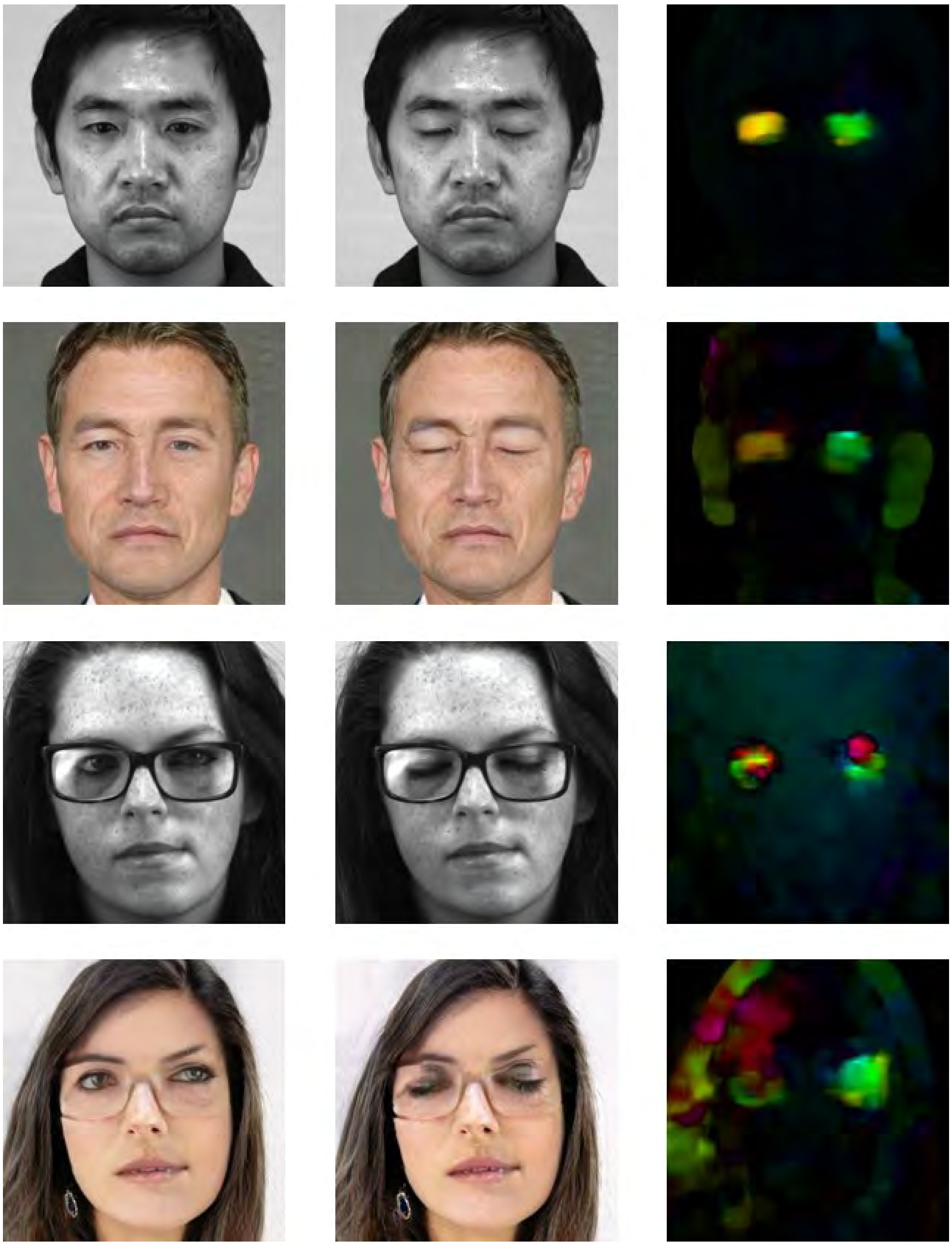


Figure 4.14: Eye blinks can cause changes in other facial regions. (**Top**) Mild case which results in small changes of hair and ears in the synthetic images. (**Bottom**) Extreme case on participant with glasses which results in huge changes.

4.3.5 Conclusions

We showed that style transfer on a pre-existing long video dataset (SAMM-LV) can be used as a method of generating a new synthetic dataset – SAMM-SYNTH. We found that AU6, AU12, and AU45 are AUs that transferred well in the SAMM-SYNTH using Pearson's correlation. We performed facial motion transfer analysis using optical flow to visualise the movements on both the original and synthetic data. Like in other GAN-based approaches, we observed the synthetic data were affected by visual artifacts.

With additional data, the ME spotting algorithm shows improvement. The advanced image augmentation using neural style transfer which acts as an advanced image augmentation that increases the variety of input data. However, there is a trade-off between ME and MaE spotting performance.

Future work includes addressing the style transfer issues related to eye blinks, eye glasses, and the visual artifacts. We will expand the training dataset of the style transfer model to include other face datasets and design a new method. To evaluate the effectiveness of synthetic data as a data augmentation technique in this domain, we will add SAMM-SYNTH to the current training pool and conduct ME related experiments to investigate the use of SAMM-SYNTH in ME recognition, and spotting ME and MaE in long videos.

4.4 Summary

SAMM-LV [31] is a long video dataset containing both facial micro- and macro-expressions. We evaluated the performance of OpenFace facial behaviour tools in AUs detection. As OpenFace was trained on macro-expression, for AU presence analysis and spotting (combined of both ME and MaE), it achieved a reasonable results with $Accuracy$ of 0.4712 and $F1 - Score$ of 0.3299 respectively. The performance dropped in micro-only videos with $Accuracy$ of 0.2903 in contrast to 0.4502 in macro-only videos which further supports our hypothesis. For spotting, our overall results outperformed the baseline result of MEGC2020 in all three performance metrics of $Precision$, $Recall$, and $F1 - Score$. This evaluation method shows that AU can be used as a rough estimation for the movement related to ME and MaE.

SAMM-LV is a significant addition towards pre-existing ME long video dataset. As of the time of the release of this dataset, there was only one dataset (CAS(ME)^2). This increases the dataset pool by a significant amount. With this new dataset, it can increase the generalisation of machine learning model and makes the model more robust to other samples. This dataset has been used in Micro-expression Grand Challenges (2020 - 2022) in ME and MaE spotting task, where the baseline methods are described in the next chapter.

We also proposed a new take on generating ME and MaE data by performing style transfer on existing dataset. It is an advanced image augmentation that theoretically can generate infinite amount of data by only changing the reference image. We perform quantitative measurement using correlation analysis using the AUs detected by OpenFace for both original and synthetic data. Optical flow is also implemented for qualitative measurement.

Chapter 5

3D-CNN for ME and MaE Spotting using Temporal Oriented Reference Frame

5.1 Introduction

Facial expression is the primary form of visual information on human emotion. It can predict a person's current state of emotion. This chapter discusses two types of facial expression, i.e., macro-expression (MaE) and micro-expression (ME), which are based on the duration of these expression lasts. MaE (also known as regular facial expression) lasts from 0.5 to 4.0s [21] and it normally has higher intensity; ME occurs in less than 0.5s and more likely to have lower intensity. ME occurs more frequently in high-stake and stressful circumstances [22], [23]. As it is an involuntary reaction, the actual emotional state of a person can be studied through analysing MEs.

For ME spotting, due to limited dataset availability, early works are based on datasets consisting of short clips containing categorised ME (i.e., SAMM [37], SMIC [29], and CASME II [36]). Spotting with clips containing ME will result in high detection rate regardless. Hence, the recently created long video datasets that contains both ME and MaE, i.e., SAMM Long Videos (SAMM-LV) [31] and CAS(ME)^2 [30], were introduced to better represent spontaneous emotion for ME and MaE spotting. They also resemble real-world scenario where ME is infrequent and may co-occurs and overlaps with MaE. This section focuses on automated spotting of MaE and ME on SAMM-LV and CAS(ME)^2.

From literature review, we found most approaches are hybrid approaches, which are artificial neural network uses handcrafted features (i.e. LBP, optical flow) as input. LBP is computed by comparing the grey level value of the centre pixel with the neighbouring pixels. If the grey level of the neighbouring pixel is higher than the centre pixel, it is assigned to 1, and set to 0 otherwise (further details in Section 3.3). Some examples of method that uses LBP as input features uses fixed grid based spatial division and ROI method [70] and merges both handcrafted (LBP based) and deep (CNN) features [71]. Optical flow computes the differences of two image frames every time when it is applied within a video sequence. There are a number of optical flow based methods [100], [102], [103], [200] that detect temporal correlation of video sequences. Some approaches [102], [108] also uses recurrent neural network (i.e. LSTM) to analyse temporal information. Both LSTM and optical flow are computationally expensive. In addition, optical flow has weaknesses such as drifting over frames [206] and is very susceptible to illumination changes [207]. We also noticed that previous attempts lack duration centred analysis. We take advantage of the major difference between ME and MaE (they occur for different duration, where ME occurs less than 0.5 s while MaE occurs in 0.5 s or longer) and propose a two-stream network with a different frame skip based on the duration differences for ME and MaE spotting.

This chapter proposes a 3D-CNN that detects movements in video sequences using a frame-by-frame approach on long video ME dataset. Prior to this contribution, a simplified task which involves movement spotting was attempted. This main method that is capable of spotting both ME and MaE simultaneously is a benchmark of SAMM-LV dataset and baseline method for Micro-Expression Grand Challenge (MEGC) in 2021 [121] and 2022. Local Contrast Normalisation (LCN) is used to normalise the contrast value of the input image which results in drastic improvements in the model performance. The model performance is compared with the submissions of MEGC2020 [182].

5.2 Main contributions

The main contributions of this chapter are:

- Our approach is the first end-to-end deep learning ME and MaE spotting method trained from scratch using long video datasets (at the time of developing our method).
- Our method uses a two-stream network with temporal oriented reference frame. The reference frames are two frame pairs corresponding to the duration difference of ME and MaE. The two-stream network also possesses shared weights to mitigate overfitting.
- The network architecture consists of only 3 convolutional layers with the capability of detecting co-occurrence of ME and MaE using a multi-label system. This method has potential to be used on lightweight devices (e.g., smartphones) in real-time.
- We implement LCN that drastically improves the overall network performance across a range of configurations and parameters. We also show that LCN is essential in our network. Our 3-layer network with LCN outperforms deeper network (i.e., 20-layer network).

5.3 Facial Movement Spotting

Before conducting the full experiment, to ensure the network is capable of detecting movements, a prototype model on facial movement spotting was created. The aim of this prototype network that is to investigate whether this network architecture can spot facial movements. This provides an easier alternative as a baseline to troubleshoot the model more efficiently. This model also trains faster that significantly reduces the training time which speeds up the optimisation process. The network architecture is identical to the full version with then only difference is the final classification is movement or no movement. This version is only trained and validated on SAMM-LV dataset.

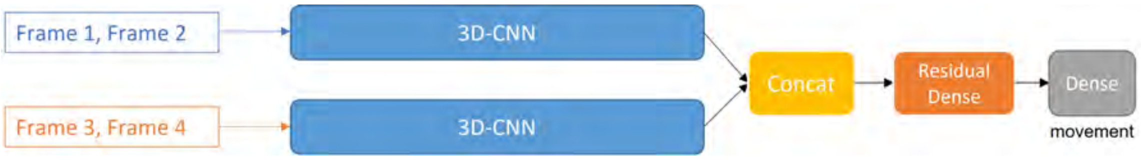


Figure 5.1: Simplified Network Architecture for Spotting Facial Movements

The training is identical to the main experiment. The input of the network is consisted of 4 images with dimension of 112×112. Leave-one-subject-out (LOSO) cross-validation and regularisation of the network are also used which will be discussed in the main experiment in

Section 5.4.4.

Weighted loss function is used to address class imbalance of the dataset where majority of the training sets are labelled neutral. This loss function is shown in Equation 5.1.

$$Loss = -\sum_{i=1}^{C'=1} W \cdot t_i \cdot log(s_i) + (1 - W)(1 - t_i) \cdot log(1 - s_i) \tag{5.1}$$

where t_i is the ground truth label, s_i is the prediction, C' is the number of class (set to 1 as we have only one class) and W is the weighing factor (set to 0.75).

The initial stage of our project, we took 3 subjects out of the training set and use it for validation. All the subjects was classified into 3 categories of low, average and high facial movement split according to the interquartile range of the frame labels. This cross-validation technique was made as it can be trained and shows the performance of the network relatively quickly. This is extremely useful for testing out different network architectures.

Through the initial investigation using this network, we find out the approximate hyperparameters and network architecture that leads to model convergence. Based on the approximated parameters and network, we proceed the full experiment using LOSO cross-validation.

5.3.1 Results and Discussion

This method is evaluated using two assessment: frame-by-frame and interval based. The raw output is frame-by-frame. Facial expression occurs within certain duration, hence, an interval based evaluation is introduced as a more suitable alternative (which is also the official evaluation metric for MEGC).

Frame-by-frame Evaluation

The results for frame-by-frame evaluation are shown in Figure 5.1. It shows that LCN plays an important role in our network whereby all metrics (loss, F1-score and AUC) are better compared to the counterpart without LCN. A deeper network also have better performance. Deeper network architectures will be studied in the future to determine the optimal network depth.

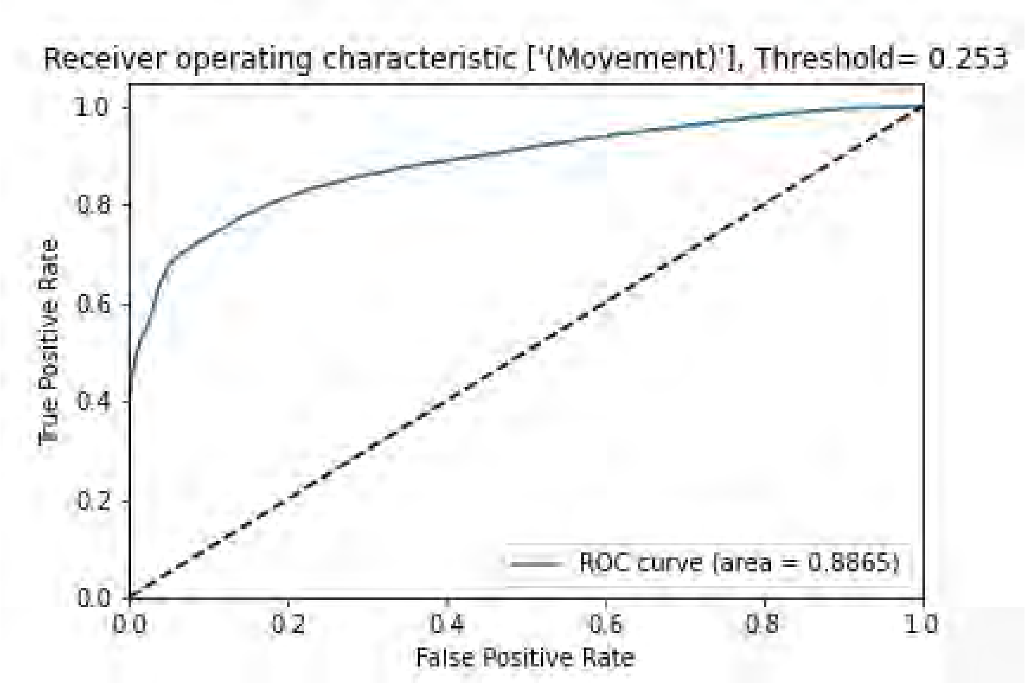


Figure 5.2: ROC curve of one selected fold.

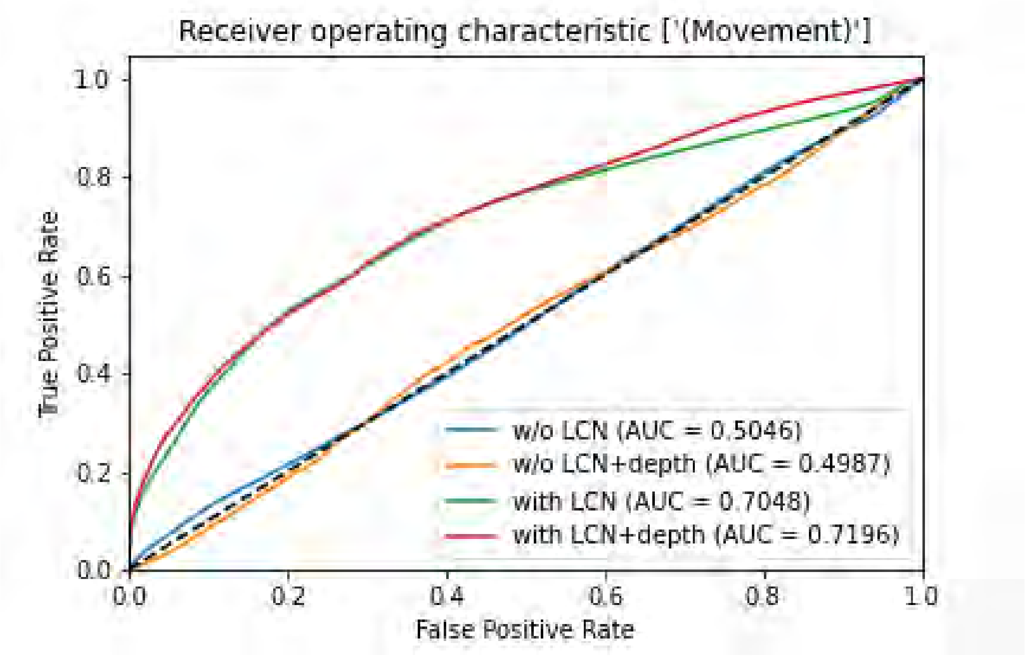


Figure 5.3: Overall ROC curve. LCN is shows results with significant better results. Without LCN, the model has performance similar to random guesses.

Intersection of Union (IoU) Evaluation

The initial finding shows that LCN and additional depth improves model performance.

Table 5.1: Frame-by-frame Evaluation for Facial Movement Spotting

Network and Parameter	**Loss**	**Precision**	**Recall**	**F1-score**	**AUC**
3D-CNN (w/o LCN)	0.3877 (±0.1498)	0.2361 (±0.1342)	0.4830(±0.1292)	0.3012 (±0.1176)	0.5046
3D-CNN (w/o LCN) + depth	0.3774 (±0.2076)	0.2724(±0.1776)	0.5103 (±0.1155)	0.3360 (±0.1569)	0.4987
3D-CNN (with LCN)	0.2518 (±0.0796)	0.3409 (±0.1871)	0.6008 (±0.1240)	0.4160 (±0.1684)	0.7048
3D-CNN (with LCN) + depth	**0.2367** (±0.0561)	**0.3599** (±0.1885)	**0.6283** (±0.1041)	**0.4387** (±0.1638)	**0.7196**

Table 5.2: Comparison using interval based IoU Methods

Network and Parameter	**Precision**	**Recall**	**F1-score**
Pan et al. [107]	0.0681	0.1010	0.0813
3D-CNN (w/o LCN)	0.0712	0.0681	0.0659
3D-CNN (w/o LCN) + depth	0.0477	0.0528	0.0491
3D-CNN (with LCN)	0.1309	0.1783	0.1464
3D-CNN (with LCN) + depth	**0.1489**	**0.2044**	**0.1660**

The initial network depth is 6 layers and additional depth increase the number of layers to 9 layers. Using LCN improves the performance significantly. However, the effect of network depth remains inconclusive, which will be explored in Section 5.4.5.

Although frame-by-frame evaluation shows better performance, facial expression is a dynamic movement of face that consists of a sequence of action. Hence, static frame evaluation has its limitation as it cannot captures the temporal changes of the expression. A more realistic assessment is interval based evaluation which is based on the overlapping between the ground truth interval and the predicted interval. These overlapping of these two intervals are compared using intersection of union. This provides a more accurate overview on the performance of the model.

5.4 Facial ME and MaE Spotting

Our goal is to detect ME and MaE within long video sequences. By using the duration difference of ME and MaE, we propose a two-stream 3D-Convolutional Neural Network (3D-CNN) with temporal oriented frame skips. We define the two "streams" as ME and MaE pathways, as illustrated in Fig. 5.6. They are structurally identical networks with shared weights, but differ in frame skips. We use 3 convolutional layers and pool all the spatial dimensions before the dense layers using global average pooling. This design constrains the network to focus on regional features, rather than global facial features. Next, we further implement the brightness and/or contrast normalisation of input images which improves model performance. This is important for generalisation and real world applications, as there is likely more variation in skin tone and brightness between different individuals, and lighting conditions. Therefore, we apply LCN to all images before presented to our network.

5.4.1 Preprocessing

Facial Alignment OpenFace 2.0 [4] is used for facial alignment. It is a general-purpose toolbox for facial analysis. Further details of this toolkit can be found in Section 3.2.3. In our experiment, image resolution is 112×112 pixels, which is the default output resolution of OpenFace.

Local Contrast Normalisation (LCN) LCN [137] was inspired by computational neuroscience models that mimic human visual perception [208] by mainly enhancing low contrast regions of images. LCN normalises the contrast of an image by conducting local subtractive and divisive normalisations [137]. It performs normalisation on local patches (per pixel basis) by comparing a central pixel value with its neighbours. The unique feature of LCN is its divisive normalisation, which consists of the maximum of local variance or the mean of global variance. If an area of image has very low variance (approximately 0), dividing with a small value will form a bright spot. Dividing using the mean of global variance mitigates this issue. The main advantage of this method is robustness towards the change in brightness or contrast (shown in Figure 5.4). The facial features are well preserved despite the random changes in brightness and contrast. This can be a solution to address the weakness of overused conventional optical flow method of dealing with uneven lighting. In our implementation, Gaussian convolutions are used to obtain the local mean and standard deviation. Gaussian convolution acts as a low pass filter which reduces noise. It also speeds up the local normalisation process as it is a separable filter (where 2-dimensional data can be calculated using 2 independent 1-dimensional functions). Further details can be found in Section 3.2.4.

The general equation of LCN can be described as

$$g(x,y) = \frac{f(x,y) - \mu_f(x,y)}{\sigma_f^2(x,y)} \tag{5.2}$$

where $f(x,y)$ is the input image, $\mu_f(x,y)$ is the local mean estimation, $\sigma_f^2(x,y)$ is the local

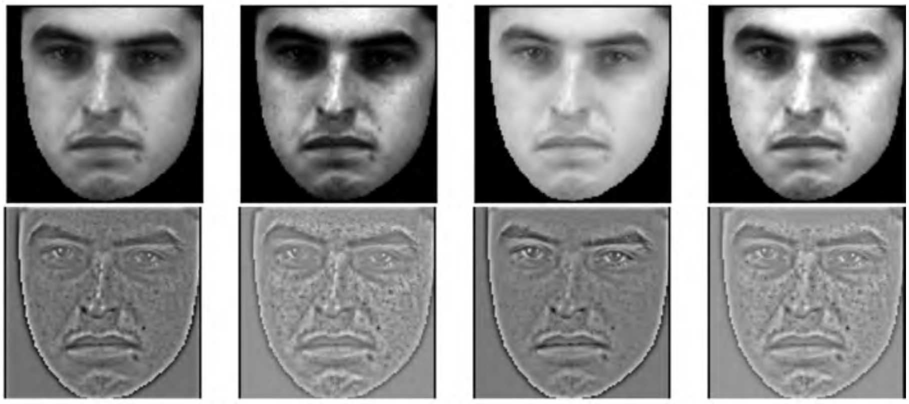


Figure 5.4: Preprocessing: (Top) Face alignment and data augmentation (randomised brightness and contrast change) on a subject of SAMM-LV; and (Bottom) Image normalised using LCN. Despite the brightness and contrast differences, the facial features remain well-preserved.

variance estimation and $g(x, y)$ is the output image.

5.4.2 Temporal Information encoded on Depth Dimension of 3D-CNN

There are multiple ways to encode temporal information such as using temporal based features (i.e. optical flow) and recurrent networks (i.e. LSTM). For this method, the depth dimension of 3D-CNN is used to capture the temporal information of the video. By using the depth dimension as time, a two-stream network is produced by avoiding convolution across the depth of the network after the first layer. The two stream network is done by striding across the depth dimension in the first convolution. This provides the differences between two frames (either long frame skip or short frame skip respectively). Each stream was then convolved independent to each other and concatenate in the end of the network. For training, a randomised frame skip (within the range of the facial expression) was used which increase the network generalisation towards ME/MaE of different duration. For testing, the temporal information encoded is through strides following k-th frame [94] (further details in Section 5.4.3).

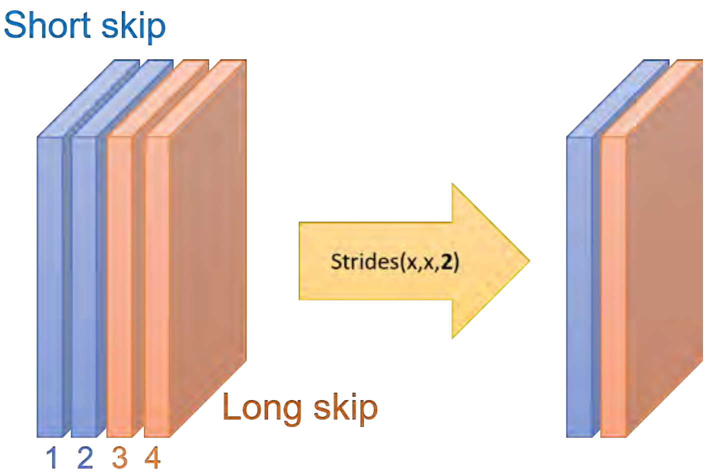


Figure 5.5: Initial stage of 3D-CNN, where x is the filter size. This is the 1^{st} layer of 3D-CNN. The temporal information is encoded in the depth dimension. The 4 images on the left are input frames. The first two frames (in blue) have shorter frame skip while the latter two have longer frame skip. The numbers represents the indices of the input frames.

5.4.3 Network Architecture

We propose a two-stream network using a 3D-CNN (network architecture shown in Figure 5.6). Our network takes advantage of the duration differences of ME and MaE and encouraging one network to be more sensitive to ME and the other to MaE. This is made possible by using a different number of skipped frames in each respective stream (using the maximum duration of a ME, 0.5s, as the threshold for the duration difference). Our network consists of depthwise separable convolutions, which has about 10% less parameters compared to regular convolution counterpart.

Input Layer The input of this network consists of 4 images. The frame pair in the first stream has a shorter frame skip compared to the latter pair. The frame skips are determined based on the k-th frame. The k-th frame, described by Moilanen et al. [94], is the average mid-point of odd-numbered facial expression interval of the whole dataset. These pairs are then fed into two separate but identical neural networks with shared weights.

Weighted loss function To the best of our knowledge, we are the first in ME spotting to weight imbalanced datasets using a loss function. The datasets used in our experiment are imbalanced, and there are more neutral frames relative to frames containing ME or MaE. We also weighted the loss based on ME and MaE, as ME occurs less than MaE. The loss can be

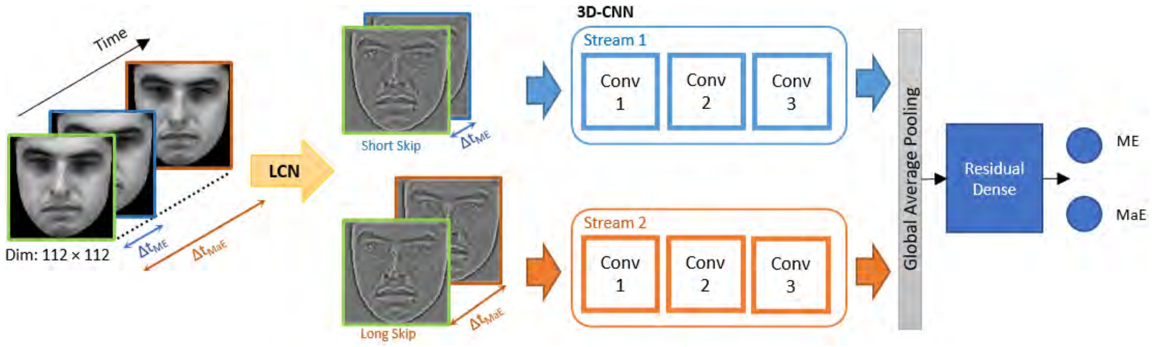


Figure 5.6: Network architecture of our two-stream 3D-CNN. It is lightweight as it has only 3 layers. Temporal oriented frame skip based on the duration differences of ME and MaE (where $\Delta t_{ME} < \Delta t_{MaE}$). Each convolutional block consists of depthwise separable convolution, batch normalisation and dropout. The residual dense layer possesses the skip connections that shares weights. Two dense nodes were used at the end to resemble the presence of ME and MaE.

described as

$$Loss = -\sum_{i=1}^{C'} M_i \cdot [W \cdot t_i \cdot log(s_i) - (1 - t_i) \cdot log(1 - s_i)] \tag{5.3}$$

where t_i is ground truth labels, s_i are the predictions, C' is the number of expression types ($C' = 2$ in our case, for ME and MaE), W is the weighting factor that functions to penalise more when the network predicts ME/MaE wrongly as neutral and M_i is the weighting factor for expression (ME or MaE).

We only apply weighted loss function when training SAMM-LV as we found out model trained with SAMM-LV improves with weighted loss function. The effects in CAS(ME)^2 is negligible. We used $C' = 2$, $M_0 = 0.9$ (for ME), $M_1 = 0.1$ (for MaE). Coefficient W used is 3. All the weighting factors are used to address the dataset imbalance. W is used to address different number of ground truth labels of ME/MaE and neutral; M_0 and M_1 is used to address the imbalanced labels of ME and MaE. The model performance of different weighting factors is shown in Table 5.10.

Depthwise Separable Convolution We use depthwise separable convolution of MobileNet [158] that reduces total trainable parameters with minimal performance impact. It consists of depthwise and pointwise convolution. Depthwise convolution is convolution applied on individual channels instead of all channel at once (as in regular convolutional). Pointwise convolution is convolution that uses a 1×1 kernel with a third dimension of d (where d is the number of channels) on the feature maps.

Table 5.3: Duration analysis of SAMM-LV and CAS(ME)^2.

Dataset	**SAMM-LV**		**CAS(ME)^2**	
Type of Expression	ME	MaE	ME	MaE
Minimum (s)	0.15	0.51	0.27	0.10
Mean (s)	0.37	2.17	0.42	1.25
Maximum (s)	0.51	25.88	0.53	3.90

GAP and Residual Dense Layer A global average pooling (GAP) layer is used to flatten the convolution output and enforce modelling of localised facial movements. It is followed by the final hidden layer consisting of a residual dense layer. This layer consists of two fully connected layers with skip connections inspired by ResNet [159].

Output Layer The output layer consists of two dense nodes representing the presence of ME and MaE. A sigmoid activation function is used as the output, and is in the range between 0 and 1.

5.4.4 Experiment

Our experiment involves an end-to-end 3D-convolutional network using leave-one-subject-out (LOSO) cross-validation. This cross-validation is the standard cross-validation method used in multiple MEGCs [8], [121], [182]. k-fold cross-validation (e.g. 5-fold or 10-fold) was not used in ME research as the available dataset is relatively small. This section provides dataset information and training details of our experiment.

Datasets

The datasets used are SAMM Long Videos (SAMM-LV) [31] with 147 long videos containing 343 MaEs and 159 MEs; and CAS(ME)^2 [30] with 87 long videos containing 300 MaEs and 57 MEs. The duration analysis of MEs and MaEs in the long videos are shown in Table 5.3. It is noted that the ME duration of SAMM-LV is shorter than CAS(ME)^2, but the MaE duration is longer. The original ground truth of these datasets consist of onset, apex, and offset frame labels of each facial expression. We label the ground truth of movement from the onset frame to the offset frame, inclusively. Our ground truth consists of two labels of binaries where “0” represents absence while “1” represents presence of ME or/and MaE.

Training

Randomised frame skips are used in training and validation. This creates a more realistic scenario as the duration of each facial expression is unknown in real life. It can also act as a regularisation process by adding variations and perturbations to the input. For model testing, we used a frame skip based on the k-th frame of ME and MaE of each respective dataset shown in Table 5.5. The visual differences of frames calculated using this interval (frames skipped) is larger, making the facial movements more distinct for the algorithm to spot.

Regularisation Random augmentations (i.e., contrast, gamma intensity, gamma gain and rotation) on the input images are performed as shown in Table 5.4. A 50% probability of horizontal flip is also included. Other regularisation include adding dropout layers and random frame skips during training and validation. These augmentations create input images with slight variations. This improves the classification performance and generalisation of the model by forcing the model to learn the differences between each frames instead of between each sets of frames.

Table 5.4: Input Image Data Augmentation

Augmentation	**Min Value**	**Max Value**
Contrast	0.5	1.5
Gamma	0.5	1.5
Rotation	-10.0°	10.0°

Training Configuration As shown in Table 5.5, the results are evaluated using leave-one-subject-out (LOSO) cross-validation. For full training process, we modify our LOSO to leave two subject out instead of one subject. Our algorithm takes two subjects (named Subject A and Subject B) out as validation and test sets while the remaining subjects are used as training set. Each training loop is initiated by taking Subject A and B out of the total dataset. Next, train on the remaining subjects and evaluate using Subject A. Save the best model with the tracked metric and if the metric does not improved for 5 consecutive epochs (track using variable "*patience*"), end the training process and test the saved model using Subject B. In our experiment, the tracked metric is the validation loss of each epoch. This process is also repeated in the same time using Subject B as validation set and Subject A as test set. The pseudo-code for LOSO used in our experiment is shown in Algorithm 1.

The main advantage of this method is the required training time is cut by half by leaving out two subject (instead of one). In our experiment, SAMM-LV has 30 subjects, it requires 15 sets of training instead of 30 sets and CAS(ME)^2 has 22 subjects, it requires 11 sets if training instead of 22 sets.

Algorithm 1: LOSO cross-validation

```
Determine the number of subjects;
Determine $num\_folds$ by dividing no. of subject with 2 (round up to the nearest integer);
for 1 : num_folds do
    Hold two subjects: named them as Subject_A and Subject_B;
    Use the rest of the subjects as training set;
    while While patience < 5 do
        Train model;
        Validate using Subject_A and Subject_B;
        if val_loss < lowest_loss then
            patience = 0;
        else
            patience = patience + 1;
        end
    end
    Load best model validated by Subject_A and test using Subject_B;
    Load best model validated by Subject_B and test using Subject_A;
    Record all the performance metrics and compare with ground truth;
end
```

The distribution of training, validation and test set for each respective folds of LOSO are shown in Figure 5.7. This shows that each fold contains different amount of training and validation/test set. This implies that the video length of each subject is not the same. This is due to SAMM-LV dataset is consisted of videos of spontaneous reaction of the participants.

Early stopping is used during training, ending when the loss does not improve for 5 consecutive epochs.

Table 5.5: Training configuration. Stream 1 is designed to be more sensitive to ME, while Stream 2 is more sensitive to MaE by using different range of frame skips based on the duration differences of ME and MaE. The k-th frame is the average mid-point of facial expression interval. Note: * used in training and validation, † used in testing

Dataset	**SAMM-LV**	**CAS(ME)^2**
Batch Size	16	
Learning Rate	0.007	0.005
Random frame skip* (Stream 1 & 2)	25∼75 & 200∼400	3∼9 & 16∼50
k-th frame skip† (Stream 1 & 2)	37 & 217	6 & 19
Manual frame skip† (Stream 1 & 2)	30 & 310	10 & 33

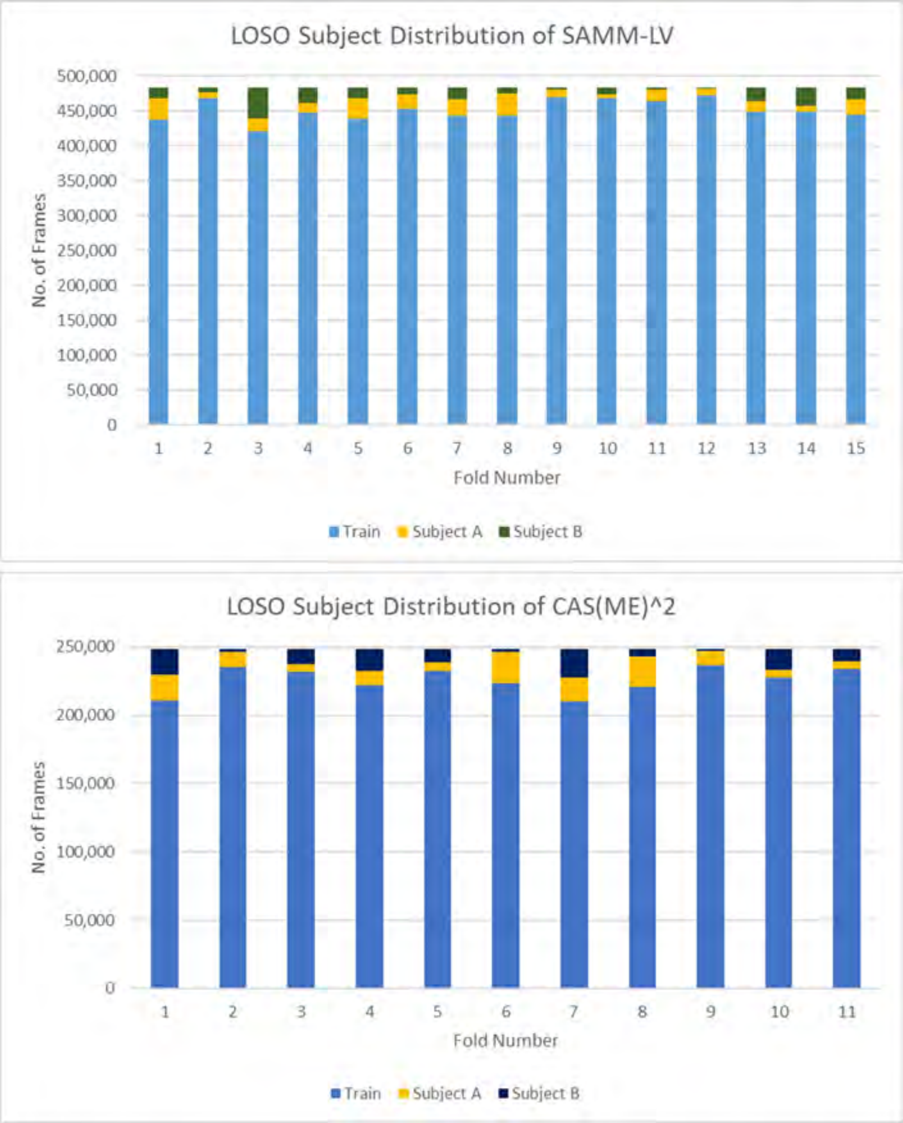


Figure 5.7: LOSO Subject Distribution of SAMM-LV and CAS(ME)^2. It shows that each fold contains different amount of training and validation/test set.

Table 5.6: Results (Raw Output) of macro- and micro-expression spotting of our method

	MaE		**ME**	
	F1-score	AUC	F1-score	AUC
SAMM-LV	0.3872	0.6780	0.0720	0.5687
CAS(ME)^2	0.1369	0.6925	0.0174	0.5762

5.4.5 Results

Our network predicts the presence of facial expression on a per frame basis. F1-score and area under the curve (AUC) of receiver operating characteristic (ROC) of our raw output are reported in Table 5.6. One selected fold for ROC curve is shown in Figure 5.8. We compare each frame using normalised results filtered using a threshold based on the ROC curve. From the F1-scores, our model performs better on SAMM-LV. However, CAS(ME)^2 performs better for the AUC. Across both dataset, the F1-score and AUC of the ROC curve of MaE is higher than ME which indicates our network is better in distinguishing MaE from neutral frames than ME counterpart. This is within expectation as MaE often has higher intensity and longer duration which makes MaE easier to spot.

We apply the Intersection over Union (IoU) method used in Micro-Expression Grand Challenge (MEGC) 2020 [182], [200] to compare with other methods. The interval is then evaluated using the following IoU method

$$\frac{Predicted \cap GT}{Predicted \cup GT} \geq J \tag{5.4}$$

where J is the minimum overlapping to be classified as true positive, GT represents the ground truth expression interval (onset-offset), *Predicted* represents the detected expression interval. In our experiment, J is set to 0.5.

As other methods use different post-processing steps, we decided to use two different evaluation methods. The first method is our Automated IoU Method and the second method is Multi-Scale Filter used by Zhang et al. [100].

Automated IoU Method

We convert our results into intervals using automated thresholding based on ROC evaluation. First, the test results are normalised and smoothed using a Butterworth filter [188], which is a

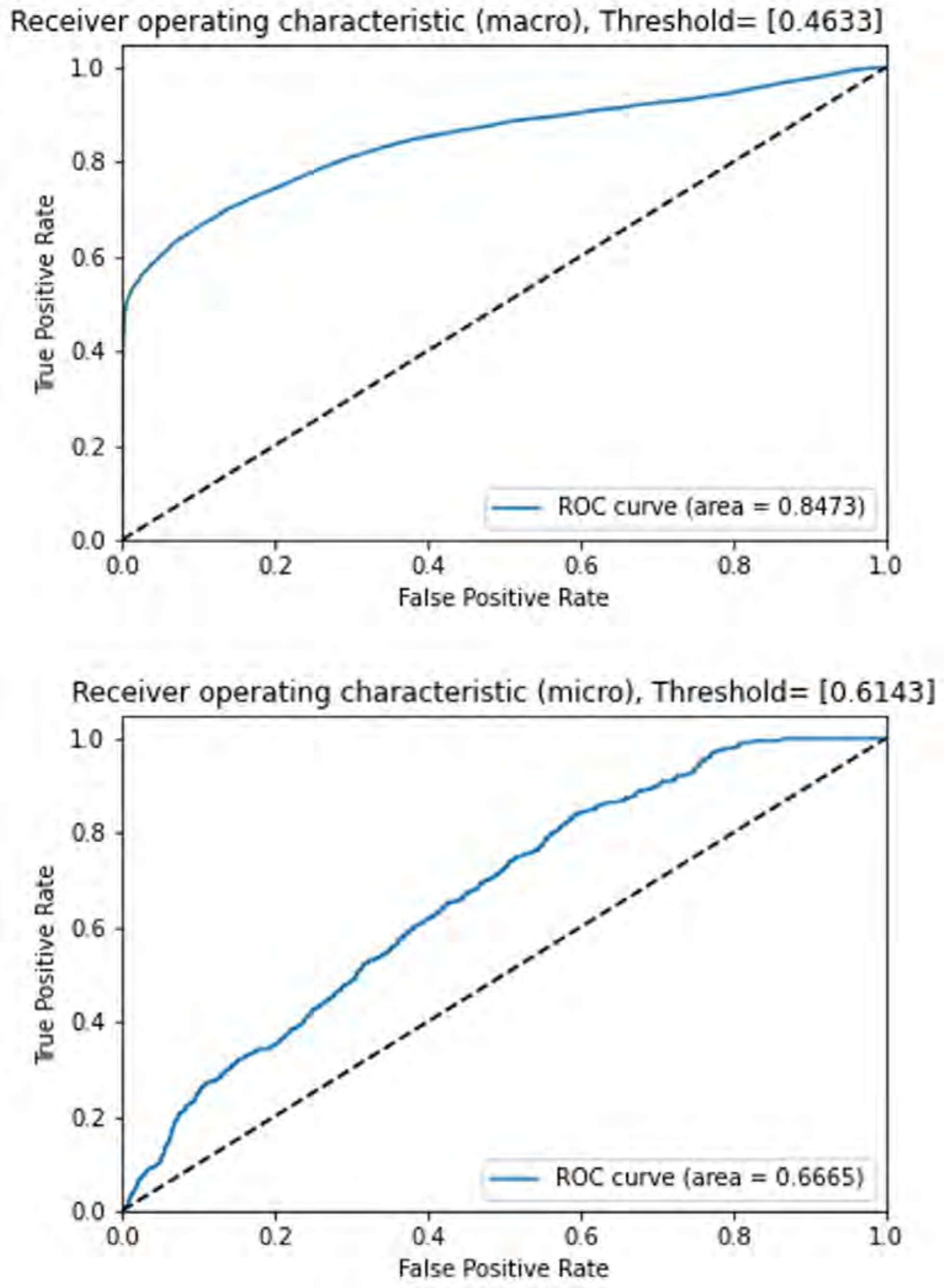


Figure 5.8: ROC curve of one selected fold for ME and MaE spotting.

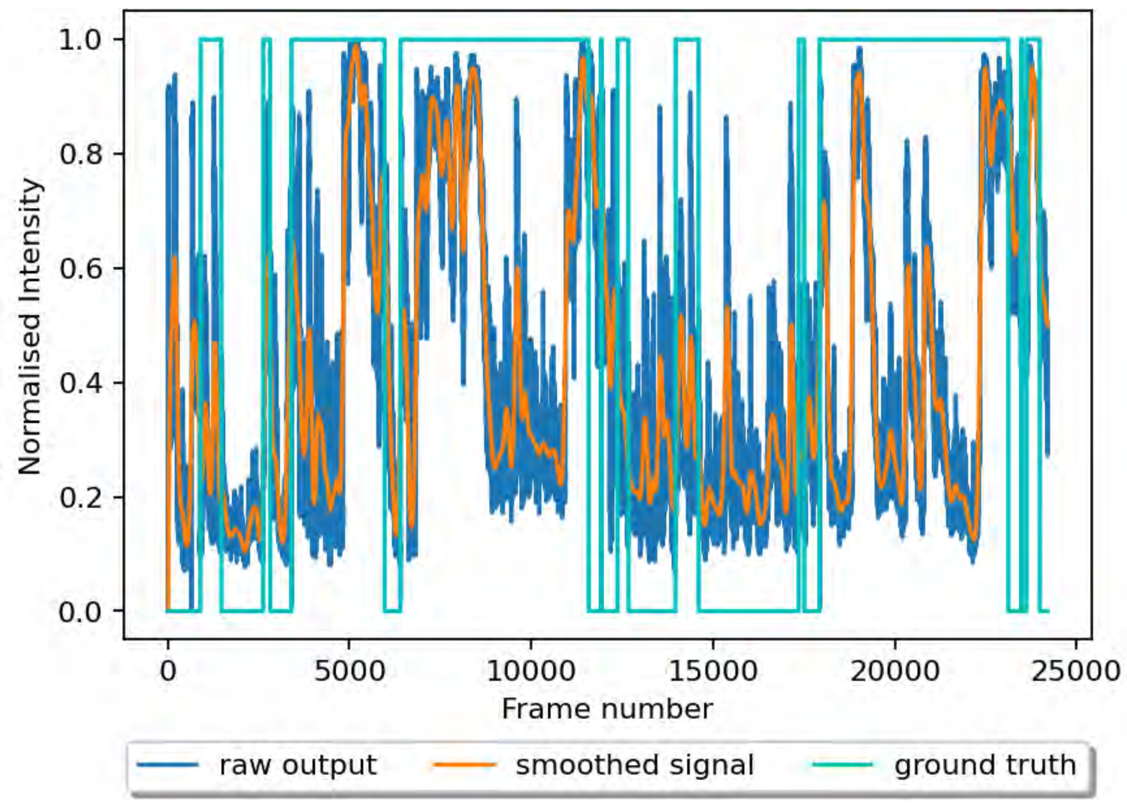


Figure 5.9: Real long video testing data of a subject smoothed using Butterworth filter with ground truth comparison

low-pass filter that cuts off high frequency noises while retaining low frequency signals, results shown in Fig. 5.9. The main advantage of this filter is it has a flat magnitude filter whereby signals with frequency below cut-off frequency do not undergo attenuation. Next, the onset and offset of both ground truth and the predictions are obtained. Finally, the overlapping was analysed using the IoU method (where TP must fulfill the criteria in Equation 3.13).

Our results show better spotting performance in SAMM-LV compared to CAS(ME)^2. One possibility is SAMM-LV has higher frame rate (200 fps) and the randomised frame skipping used in our training pipeline has more variety of input data to be learnt compared to CAS(ME)^2 (30 fps). Hence, our model is able to learn data with more variation in SAMM-LV and show better performance. ME which occur in less than 0.5s, has a small window of detection. A lower ME detection rate in CAS(ME)^2 might also be a consequence of the lower frame rate.

Comparison with the state of the art

Zhang et al. [100] and He et al. [200] are conventional approaches. These methods use post-processing steps to enhance ME spotting rate. Hence, it is not a fair comparison with

Table 5.7: F1-score of ME and MaE spotting using our Automated IoU Method, where Ours* represents our proposed method with k-th frame skip and Ours** represents our proposed method with manual frame skip. Manual frame skip is performed by first taking k-th frame as a reference, proceeded by increasing or decreasing the frame skips until the results improve.

Method	**SAMM-LV**			**CAS(ME)^2**		
	MaE	ME	Overall	MaE	ME	Overall
Pan [107]	-	-	0.0813	-	-	0.0595
Ours*	0.1504	0.0421	0.1017	0.0704	0.0075	0.0509
Ours**	**0.1543**	**0.0442**	**0.1050**	**0.0874**	0.0075	**0.0630**

our method. Instead, we use Zhang et al.'s post-processing steps (also named Multi-Scale Filter [100]) and the results are shown in Table 5.8. This method uses Savitzky-Golay filter for noise removal and signal merging as described in Zhang et al.'s paper. We obtained a notable improvement in ME and MaE spotting, particularly in CAS(ME)^2. By implementing these post-processing steps, our method outperforms in SAMM-LV and CAS(ME)^2 in MaE spotting and overall performance. Although we obtained better results using this evaluation, we noticed that this method requires selection of hyperparameters (e.g., window size and order of Savitzky-Golay filter, the upper limit of interval distance to merge etc).

For the purpose of comparison with benchmark algorithms, we implemented this method. However, we will not recommend these post-processing steps as each hyperparameter can be customised to improve the results, which might result in overfitting. As stated in Zhang et al.'s paper: "the results are terrible" before the post-processing steps. In contrast, our proposed method is already competitive before these post-processing steps, as shown in Table 5.7. Overall, our method performed the best on SAMM-LV without post-processing steps, with an F1-score of 0.105. With post-processing steps, our method achieved the best F1-Score of 0.1675 on CAS(ME)^2. It is noted that our method achieved the best result in MaE spotting on both datasets.

Visualisation using Grad-CAM

We visualise the activation of our network using Gradient-weighted Class Activation Mapping (Grad-CAM) [189]. This provides interpretable visualisation on the face region that the network is focusing on when spotting ME/MaE. We select the deepest interpretable layer, which is the last dense layer, and visualise its activation on SAMM-LV participants.

Table 5.8: F1-score of ME and MaE spotting using Multi-Scale Filter (manual post-processing steps used by Zhang et al. [100]), where Ours* represents our proposed method with k-th frame skip and Ours** represents our proposed method with manual frame skip. This post-processing steps involves signal smoothing using Savitzky-Golay filter and signal merging when intervals are close to each other.

Method	**SAMM-LV**			**CAS(ME)^2**		
	MaE	ME	Overall	MaE	ME	Overall
He [200]	0.0629	0.0364	0.0445	0.1196	0.0082	0.0376
Zhang [100]	0.0725	**0.1331**	0.0999	0.2131	0.0547	0.1403
Ours*	0.1569	0.0512	0.1083	0.1880	0.0583	0.1449
Ours**	**0.1595**	0.0466	**0.1084**	**0.2145**	**0.0714**	**0.1675**

In Figure 5.10, we observe that the heatmaps closely resemble the Facial Action Units (AUs) of facial expression. The reliable AUs of happiness are associated with AU6 (Cheek Raiser) and AU12 (Lip Corner Puller). Figure 5.10 (a) illustrates the heatmap of happiness, where it shows high activation around eye corner (AU6) and the mouth region is directed to the upper part of the face (AU12). On the other hand, the reliable AUs of sadness are AU4 (Brow Lowerer) and AU15 (Lip Corner Depressor). In Figure 5.10 (b), the heatmap shows activation on both the brows and the eyes region that indicates AU4, and the mouth region forms a huge inverted curve extending until the bottom of the face which resembles AU15.

Ablation Studies

Table 5.9 shows the ablation studies conducted using our Automated IoU Method. With manual frame skip, our proposed method achieves the best result. Our proposed approach (with k-th frame skip) is not far behind, which shows that it is a viable method. We conduct our experiment without LCN using architecture with different depth, i.e., 3, 6, 10 and 20 layers. Even with a deeper network, without LCN, it performs worse than our proposed model, which indicates that LCN is a crucial component of our approach. We also replace the GAP layer with Global Max Pooling. Although both average and max pooling, average pooling identifies all discriminative regions more completely [190]. As ME is a subtle facial movement, detailed oriented average pooling is more suitable. This is further concluded in the performance of Global Max Pooling compared with our proposed network that uses GAP. Our proposed model performs worse without batch normalisation. We showed that each component of our network are essential and has a positive contribution to the overall performance.

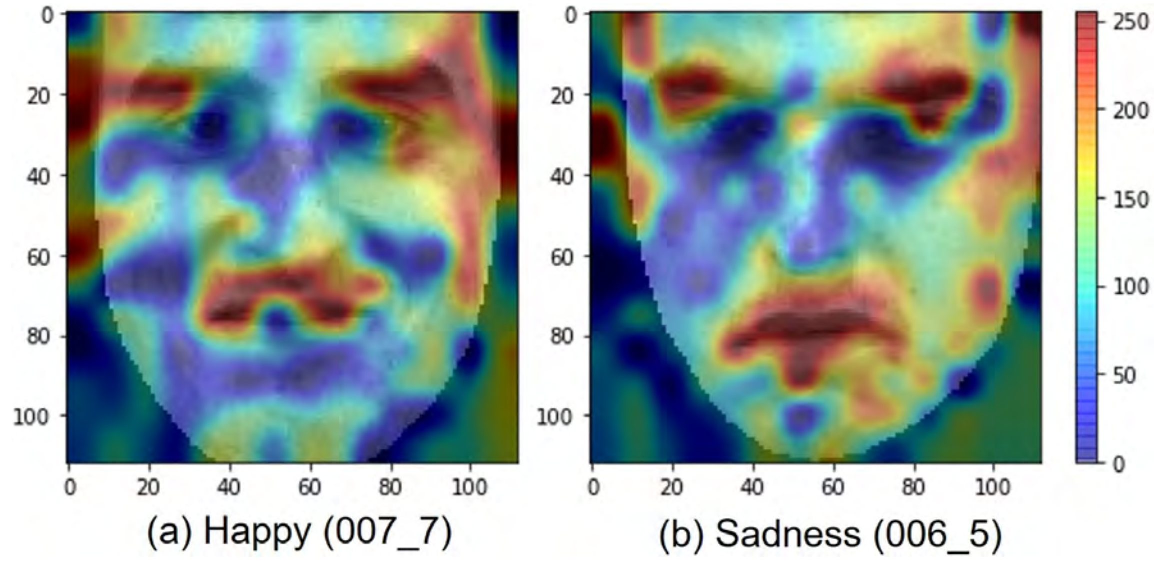


Figure 5.10: Grad-CAM visualisation on SAMM-LV participants. In (a), AU6 (Cheek Raiser) and AU12 (Lip Corner Puller) is detected; In (b), AU4 (Brow Lowerer) and AU15 (Lip Corner Depressor) is detected. AU12 and AU15 are distinctly distinguished: in (a), the mouth region heatmap is directed towards the upper part of the face; in (b), the heatmap at the mouth region forms a huge inverted curve extending towards the bottom of the face.

We conducted ablation studies on the weighted loss function (as described in Section 5.4.3). The weighted loss function is used to address the training dataset imbalance. The results are shown in Table 5.10. We only apply weighted loss in the model trained on SAMM-LV as it shows no significant improvement in the model trained on CAS(ME)^2. We weight ME more with respect to MaE by setting M_{ME} to 0.9 and M_{MaE} to 0.1; and W is used to impose a harsher penalty when the network predicts ME/MaE as neutral wrongly. We can see that without weighted loss, the network performs worse. We also demonstrate fine-tuning of W and the setting of W as 3 achieves the best performance.

5.4.6 Discussion

Model comparison Our model is the state-of-the-art in SAMM-LV and competitive in CAS(ME)^2. Conventionally, optical flow methods have good performance but require extensive preprocessing and post-processing steps, which are computationally expensive. He et al. [200] and Zhang et al. [100] use image segmentation or ROI selection, followed by optical flow extraction and spatio-temporal fusion of each ROI. On the contrary, our method is an end-to-end solution with 3 layers of CNN.

Table 5.9: Ablation studies on our Automated IoU Method: F1-scores reported. Manual frame skip fine-tuning can produce slightly better results. With LCN removed, our network performance dropped. Even with a deeper network (i.e., 20 convolution layers), it still under performs when compared to our proposed 3-layers deep network. The model performance dropped without batch normalisation and when GAP is replaced with Global Max Pooling.

	SAMM-LV			**CAS(ME)^2**		
	MaE	ME	Overall	MaE	ME	Overall
Without LCN						
- 3 Conv Layers	0.0297	0.0066	0.0198	0.0000	0.0000	0.0000
- 6 Conv Layers	0.1079	0.0275	0.0750	0.0041	0.0000	0.0030
- 10 Conv Layers	0.0825	0.0500	0.0518	0.0098	0.0000	0.0073
- 20 Conv Layers	0.0943	0.0160	0.0646	0.0000	0.0000	0.0000
Replace GAP with GlobalMaxPool	0.1311	0.0149	0.0795	0.0173	**0.0098**	0.0153
Without BatchNorm	0.1456	0.0252	0.0934	0.0510	0.0059	0.0359
Proposed	0.1504	0.0421	0.1017	0.0704	0.0075	0.0509
Manual Frame Skip	**0.1543**	**0.0442**	**0.1050**	**0.0874**	0.0075	**0.0630**

Table 5.10: Ablation studies on different weighting coefficients trained on SAMM-LV. For SAMM-LV, weighted loss function improves the detection rate. We did not report on CAS(ME)^2 dataset as we found that the weighted loss function has minimal effect on model performance.

	W	$\mathbf{M_{ME}}$	$\mathbf{M_{MaE}}$	**MaE**	**ME**	**Overall**
Proposed	3	0.9	0.1	**0.1504**	**0.0421**	**0.1017**
W/o Weighted Loss	1	1.0	1.0	0.1480	0.0238	0.0910
W/o Weighted M	3	1.0	1.0	0.1404	0.0099	0.0594
W/o Weighted W	1	0.9	0.1	0.1413	0.0268	0.0900
Vary coefficient W	6	0.9	0.1	0.1443	0.0213	0.0888
Vary coefficient W	10	0.9	0.1	0.1302	0.0240	0.0825
Vary coefficient W	0.5	0.9	0.1	0.1339	0.0262	0.0696

Zhang et al. [100] is the only method that did poorly in spotting MaE compared to other categories in SAMM-LV. Commonly, MaE (regular facial expressions) is easier to detect when compared to ME. As SAMM-LV is a dataset with high frame-rate of 200 fps, Zhang et al.'s optical flow on consecutive frames approach is unable to capture the long range dependency of MaE, which explains their relative poor results on MaE of SAMM-LV. Zhang et al. [100] is also the only method that shows better performance in CAS(ME)^2 than SAMM-LV. This may imply that Zhang et al. is heavily biased towards CAS(ME)^2. Another possible reason is the post-processing method may be more suitable in CAS(ME)^2. Our method using Zhang et al.'s post-processing has also shown notable improvement in CAS(ME)^2. The merging process in Zhang et al's post-processing is questionable and can be a potential source of overfitting. For example, 3 false positives can be merged into 1 true positive, which greatly improves the results (as shown in [100]). Moreover, this method cannot improve with additional data, whereas ours is expected to improve [209]. We provide an important contribution, justifying collection of further data.

To date, Pan et al. [107] is the only deep learning approach for spotting MaE and ME in long videos, evaluated using IoU method of MEGC2020 [182]. Comparing with this method, our model with manual frame skipping has better performance in both datasets. We also produce a complete report on all three spotting categories. Our method is able to spot ME, MaE and co-occurrence of both types of facial expression, which are the features absent in [107].

***k*-th frame skip** We investigate the effectiveness of k-th frame used by manually vary the frame skips. By varying the frame skips by taking k-th frame as initial reference, the results show only slight improvement. This indicates that k-th frame method remains a good measurement for frame skip.

Automated vs manual method Both our Automated IoU Method (automated method) and Multi-Scale Filter (manual method) show similar performance on model trained on SAMM-LV. This shows that our Automated IoU evaluation works well on SAMM-LV with only a minimal performance increment via manual method. However, it is not the case for model trained on CAS(ME)^2. The disparity of the performance on CAS(ME)^2 in both methods might be a result of different noise removal method used (Butterworth filter and Savitzky-Golay filter). The automated Butterworth filter is not adaptive enough in handling different noises, whereas using Savitzky-Golay, we can decide a suitable window size and order of filter for each respective noise. Despite higher performance detected in Savitzky-Golay, in real-world applications, automation is preferred as it is not realistic to fine-tune hyperparameters when we make prediction. With further refinement, our proposed Automated IoU Method has potential for real-world applications.

5.5 Trained composite data using SAMM-LV and SAMM-SYNTH

Table 5.11: F1-score of ME and MaE spotting using our Automated IoU Method with k-th frame skip. All results are based on LOSO cross-validation.

Method	**Results**		
	MaE	ME	Overall
Pan [107]	-	-	0.0813
Ours (results from Chapter 5)	**0.1504**	0.0421	0.1017
Train with SAMM-LV & SAMM-SYNTH, evaluate with SAMM-LV	0.1373	**0.0513**	**0.1036**
Train with SAMM-LV & SAMM-SYNTH, evaluate with SAMM-SYNTH	0.1465	0.0221	0.0955

To evaluate the effect of addition of generated data in the training set, we re-train our model using composite dataset consisted of SAMM-LV and SAMM-SYNTH. The synthetic data acts as an advanced image augmentation which doubles the data pool. The results show that with additional data, ME spotting performance improved. There is quantitative evidence [210] that training artificial neural network with more data will improve the performance of model. With more diverse data, the model is presented with more feature to learn and this can improve the model performance on unseen data.

Additional data theoretically increases input variety of the model. For our case, it improves ME and overall spotting performance but shows a drop in MaE spotting performance. A possible reason for the drop in MaE spotting performance is the generation artifacts. As the frame skip of the MaE stream is larger, the frame difference when artifacts formed is more distinct compare to lower frame skip (of the stream that uses k-th frame of ME). The results of evaluation with SAMM-SYNTH shows a drop in performance across both ME and MaE. Style transfer is a type of image augmentation and evaluating on augmented data that contains some innate randomness lowers the evaluation results.

5.6 Baseline Result for MEGC2022

In MEGC2022, the evaluation method is the same with previous MEGC which is interval based using IoU overlapping (further details in Section 3.5.3). The main difference between this challenge compared to previous challenges is the evaluation is standardise by using Grand Challenge website (url: `https://grand-challenge.org`) which makes the evaluation fairer to all participants.

The test set is consisted of SAMM Challenge Dataset and CAS(ME)^3 [211] evaluated separately. SAMM Challenge (200 fps) is consisted of 5 videos chosen from unused video of SAMM dataset. 5 videos of CAS(ME)^3 (30 fps) are selected for the challenge. Although CAS(ME)^3 is dataset that has an additional depth information, only the frontal view was used in this challenge. All videos are consisted of front-facing spontaneous reaction of participants viewing emotion eliciting videos. Similar to the other pre-existing dataset, the ground truth are AUs labelled by human coders.

For evaluation on SAMM Challenge, we train our network using SAMM-LV; for CAS(ME)^3, the network was trained on CAS(ME)^2. We achieved an overall F1-score of 0.1351, with 0.1176 and 0.1739 on SAMM Challenge Dataset and CAS(ME)^3, respectively. It is noted that on unseen dataset, our method performed better in detecting ME of CAS(ME)^3. Table 5.12 shows the baseline result to facilitate MEGC2022 Spotting Task.

Table 5.12: F1-score of ME and MaE spotting on unseen test set of MEGC2022 that uses SAMM Challenge and CAS(ME)^3

Method	**SAMM Challenge**			**CAS(ME)^3**			**Overall**
	MaE	ME	Overall	MaE	ME	Overall	
Ours	0.1739	0.0714	0.1176	0.1622	0.2222	0.1739	0.1351

5.7 Summary

We presented a temporal oriented two-stream 3D-CNN model that shows promising results in ME and MaE spotting in long video sequences. Our method took advantage of the duration difference of ME and MaE by making a two-stream network that is sensitive to each expression type. Despite only having 3 convolutional layers, our model showed state-of-the-art performance in SAMM-LV and remained competitive in CAS(ME)^2. LCN has proven to have significant improvement in our model and the ability to address uneven illumination, which is a major weakness of optical flow. LCN is also crucial for model performance in both movement spotting and ME/MaE spotting. We demonstrated our 3-layer network with LCN outperforms deep network with 20 convolutional layers. Further improvements include embedding facial landmark detection into the algorithm and simplifying the spotting algorithm to allocate more computational resources for real-time ME analysis. By adding data generated from Section 4.3 to the training set, the ME spotting performance shows improvement. We also attempted a fully end-to-end pipeline on composite dataset for our current architecture. However, it failed to detect ME and with low MaE detection rate which is discussed in the Appendix B.

Chapter 6

Generative Methods

6.1 Introduction

Generative models (e.g. GAN [172] and VAE [212]) in general generate synthetic data using input sampled from a random distribution. With a randomised input, theoretically these models are able to generate an infinite amount of new data with minimal supervision which makes them a powerful tool for data creation.

Due to the subtlety and rarity of the occurrence of ME, realistic ME generation remains unsolved. ME generation is still in an early stage at the time of writing. In current research stage, majority of the approaches either transfer movements onto the another face or uses guided variables to manually alter facial movements using existing models [110], [117] as an alternative of generating more data. Xie et al. [125] is the only novel ME generation method based on facial AU transfer between existing datasets (CASME II [36] and SAMM [37]).

The main goal of this chapter is to address data deficiency in current research field. To achieve this aim, we applied computational solutions to generate more spontaneous data based on existing data. Our solution is a recursive MaE sequence generation model that uses two frames for initialisation. We also show that the MaE generation model can be applied in ME generation task by selecting the training dataset sampled around the apex frame.

6.2 Facial Expression Generation using only Latent Representation

Video sequence generation and anticipation is one of the crucial components for estimating likelihood of future events. Potential applications include self-driving cars [213], [214], human fall prediction and data augmentation for sequence-based datasets. Although the models are made available to the public and could be utilised by the public, the networks are usually huge and requires powerful and the state-of-the-art like GPU machine or supercomputers to run them.

In facial expressions generation, most of the existing works focuses on single image generation [181], [215], [216]. Inspired by Pumarola et al. [110], researchers begun to generate facial expression sequences by using a linear variable to control a certain part of the face [112], [115]. However, as proven by our empirical experiments, facial expression movements are non-linear, where using a linear variable could not produce realistic expressions. Using a linear variable is an over-simplification on how facial movement works in real life. Linearity does not account for realism of the generated facial expression.

We propose a neural network-based method which uses Gaussian noise to model spontaneity in the generation process, removing the need for manual control of conditional generation variables. This model takes two sequential images as input, with additive noise, and produces the next image in the sequence. The output then replaces one of the input images (t-1), and this is done recursively to generate a full sequence with spontaneous expressions. The training process is not supervised by any facial feature extraction. Despite the lack of external input guidance, we show that realistic facial expression sequences can be generated end-to-end without facial alignment or facial landmarks detection. The model is trained using a non-guided approach without the use of labels. This way of training mitigates AU labelling errors that is dependent on other methods. Bypassing the need for ground truth labels can prevent the network from registering errors of other models.

When we train our network with single expressions, our model is capable of generating unique facial expression sequences. When our proposed network is trained with mixed facial expressions, our model is able to generate fully spontaneous expression sequences. We compared our method to current leading generation methods on a variety of publicly available datasets. Initial qualitative results show our method produces visually more realistic expressions and facial action unit (AU) trajectories. Initial quantitative results using image quality metrics (SSIM and NIQE) show the quality of our generated images is higher.

6.2.1 Naive Model

To investigate the ability to generate image sequences via recursive generation, a simplified network is built as the backbone architecture. This simplified network is trained using MNIST dataset which consisted of simple features. This model generates the next digit sequence (loop sequence of 0 to 9) based on the current observation without any recurrent mechanism. For this naive approach, the next digit sequence is generated using only single latent representation. The network architecture is shown in Figure 6.1. The recursive generation method (Figure 6.3) enables our model to capture temporal information by imposing a reconstruction loss (mean squared error) onto the next generated frame. 3 frames (2 initial frames and 1 targeted frame) is used for training. The shape of each digit is distinct and hence a simple 3 frames training is sufficient.

The possible reasons that one latent representation is sufficient for the model to generalise are:

1. MNIST digits consist of simple features which have low complexity which can be easily encoded and decoded.
2. MNIST digits are distinctly different from each other which lowers the difficulty for the model to pick up the key latent features of each digit.

This model only works on dataset with simple features. The generated images are slightly blurry which is a common limitation of autoencoder-based network as there is information loss during the downsampling of the encoder. It provides a backbone architecture for more complex data like facial expression sequences in the following section.

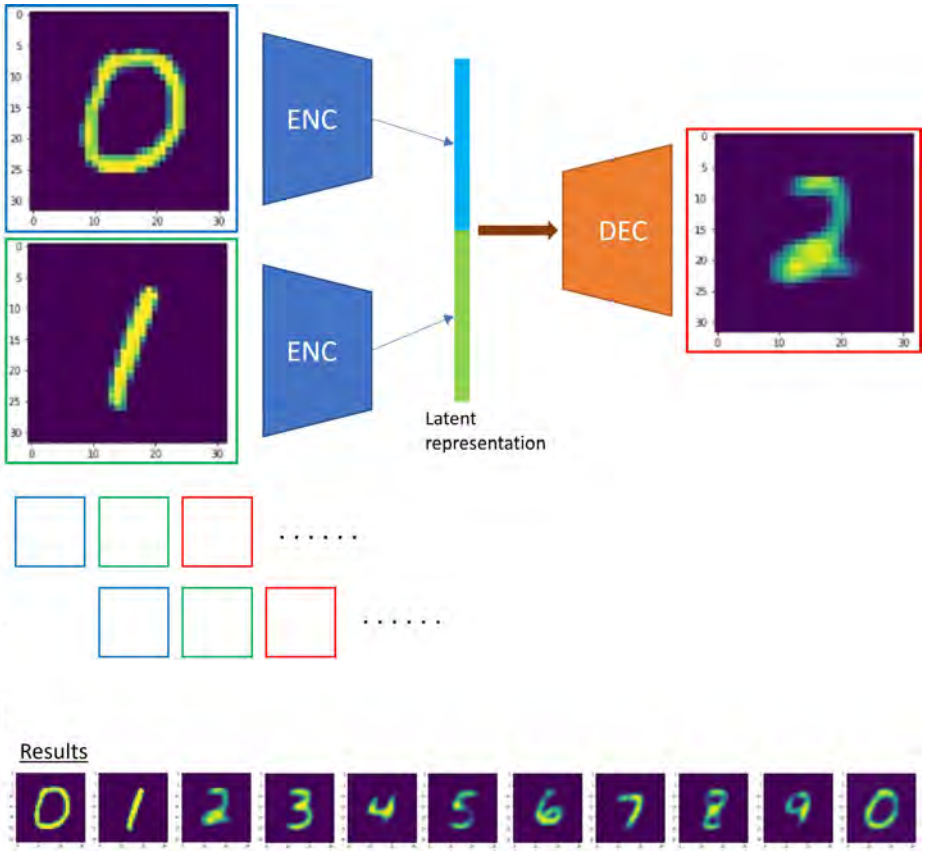


Figure 6.1: Naive approach for MNIST sequence generation. Uses one encoder to extract latent representations of frame 1 and frame 2; one decoder to generate the next frame based on the latent representations. The entire sequence is generated recursively based on the previously generated frames.

6.2.2 Main Method

There are abundance of huge models that use extensive computing power for video based generation [217], [218]. These models are impractical and impossible to be implemented by regular users due to the computing costs.

Our model is a sequential based autoencoder, which leverage the latent representation extracted by the encoder to generate subsequent image sequence. Our method was inspired by Taylor et al. [219] which is a conditional Restricted Boltzmann Machine where the generation of the next sequence is done by an undirected model with binary latent variables connected to visible variables.

The network architecture is shown in Figure 6.2. We obtain the latent representations for every downsampling convolution layers (encoder layers) and fed them into the upsampling layers (decoder layers), inspired by U-Net [220]. Typically, autoencoder based generation produces blurry images. To overcome this issue, we extract each levels of latent representation as the encoder down-samples the images. These latent representations are then fed to the decoder, which results in detailed images generation. The latent representations contain high and low level information can help in generating images with higher complexity. We found that this step is required in facial expression generation as it is a high complexity task.

Temporal information is captured using our recursive way of training network where the output frame was fed into our network (as the next "Input 2") for the subsequent generation. By comparing more than one frame, the model is able to figure out the temporal correlation between the sequence by looking a few steps further into the future.

The loss function used in this model are as follows:

$$
\begin{aligned}
P_1 &= D(E(F_c), E(F_1 + N_1), E(F_2 + N_2)) \\
P_2 &= D(E(F_c), E(F_2 + N_2), E(P_1 + N_1)) \\
P_{n>2} &= D(E(F_c), E(P_{n-2} + N_{n-2}), E(P_{n-1} + N_{n-1})) \\
loss &= \frac{1}{n}\sum_{i=1}^{i=n}(P_i - F_{i+2})^2
\end{aligned}
$$

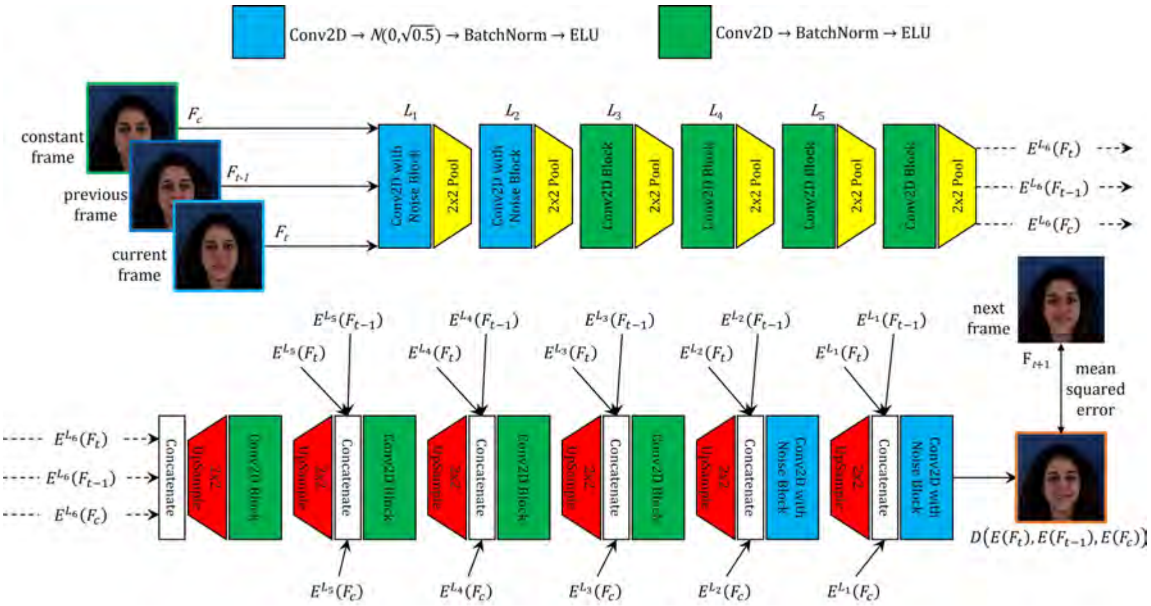


Figure 6.2: Our network architecture that takes three input frames (F_c, F_{t-1}, F_t) and predicts the next frame using auto-regression. Auto-regression of this model is performed by training our network recursively (refer to Figure 6.3). The top part is the encoder while the bottom part is the decoder. The latent representation, L , extracted by each downsampling are fed into the decoder layers. Gaussian noise is used in the shallow layers.

where F is frame originated from dataset, N is random Gaussian noise, P is generated frame, n is the number of subsequent frame used (for training only, our experiment uses n=3), i is the recursive loop number. During training, we initialise using F_1, and F_2, sampled from any part of the video (with at least 3 subsequent frames) for training. F_c is a constant neutral face that remains the same throughout the generation process. The generated frame then replaces current frame and current frame becomes previous frame as shown in Figure 6.3. Gaussian noise added in each loop primarily increases the variability of the generated frame. Each generation loop is a Markov chain. Even though the recursive loop do not contain any memory, Markov property enables model convergence for sequential generation. Equation below shows that the convergence of sequence is possible with sufficient repetition. Each recursive generation is a Markov chain. With the addition of noise, the model is still able to converge as Markov property converges the generation to a certain value.

$$M(x_{n+1}|x_1, x_2, ..., x_n) = M(x_{n+1}|x_n) \tag{6.1}$$

where M is a function with Markov property, x is the input and n is the number of repetition.

Assuming x is probability of data point, θ is model parameter and f is a certain function. The probability of X is defined as $p(x, \theta)$ as below:

$$p(x, \theta) = \frac{1}{Z(\theta)} f(x, \theta) \tag{6.2}$$

The partition function of $f(x, \theta)$, Z is defined as:

$$Z(\theta) = \int f(x, \theta) dx \tag{6.3}$$

The model parameters can be learned by minimising the negative log-likelihood of probability, $p(x, \theta)$. This negative log-likelihood is also known as the energy function, $E(x, \theta)$.

$$E(x, \theta) = -log \; (\frac{1}{Z(\theta)} f(x, \theta)) = log \; Z(\theta) - \frac{1}{K} \sum_{i=1}^{K} log \; f(x_i, \theta) \tag{6.4}$$

The derivative of contrastive divergence [151] (adapted from Woodford [221]) contains only the information of current and one of any previous state. In our implementation, we use one previous step. This is the main advantage of this method whereby energy function of the sequence can be minimised using only current and one previous state.

$$\frac{\partial E(x, \theta)}{\partial \theta} = \langle \frac{\partial log(f(x, \theta))}{\partial \theta} \rangle_{x_0} - \langle \frac{\partial log(f(x, \theta))}{\partial \theta} \rangle_{x_1} \tag{6.5}$$

Hence, by calculating the derivative of contrastive divergence, the model parameters can be learn as in the following equation where t is the time step.

$$\theta_{t+1} = \theta_t + \eta(\langle \frac{\partial log(f(x, \theta))}{\partial \theta} \rangle_{x_0} - \langle \frac{\partial log(f(x, \theta))}{\partial \theta} \rangle_{x_1}) \tag{6.6}$$

where η is the learning rate. The model parameter can be found using only one step previous and current step of observation until the derivative of contrastive divergence is converged (equal to zero). Theoretically, the model will converge after n cycles without the need of any previous memory or recurrent mechanism. We incorporated Gaussian noise on the weights of the shallow layers of our model. The addition of noise addresses overfitting and adds variability to our output data.

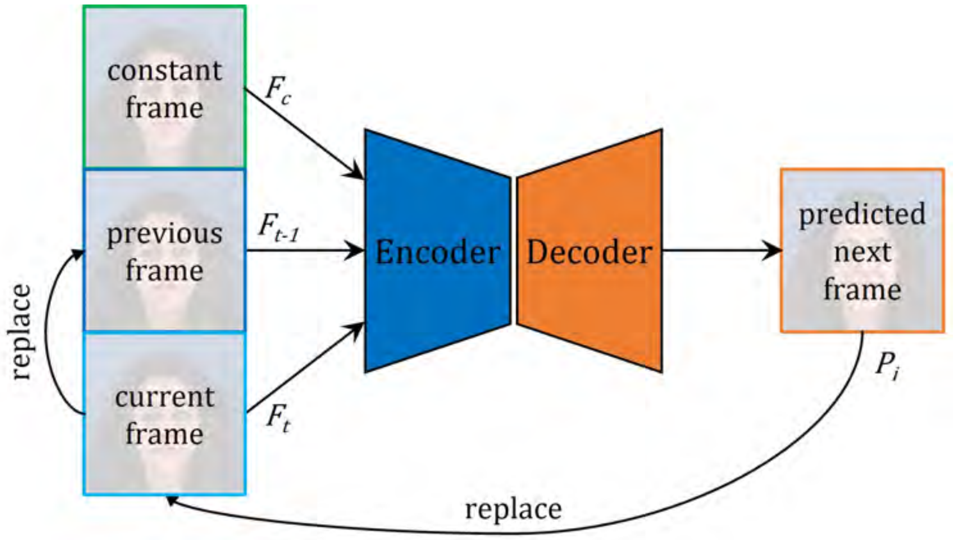


Figure 6.3: Recursive generation loop. This loop feeds the output as input for the next iteration. This arrangement allows the generation of the next sequence by replacing the output frame as current frame and current frame as previous frame in the next iteration.

Datasets

We use MUG dataset [41] as our training set. The image sequences in this dataset are consisted of front facing faces. There are six basic expressions (anger, disgust, fear, happiness, sadness and surprise). Neutral face sequences of each subjects were also captured. The frame rate of this dataset is 19 fps. The image resolution is 896×896 pixels. For model training and evaluation, we only uses the six basic expression.

We conduct facial expression generation experiments on MMI [42] and FFHQ [40] datasets. MMI dataset consists of posed facial expression sequences. FFHQ is static face images. We use FFHQ to test our model's ability in generating expression from static images.

Evaluation Metrics

We evaluate the results of the generated facial expressions sequences on realism analysis by using AU comparison and image quality assessment metrics.

AU comparison involves comparison between the AU extracted by OpenFace for real and generated sequences. Facial action unit (AU) [26] is an organised system to describe human facial movements. We compare normalised AU intensity of actual and generated facial expression across a certain time period to check the realism of generated expression.

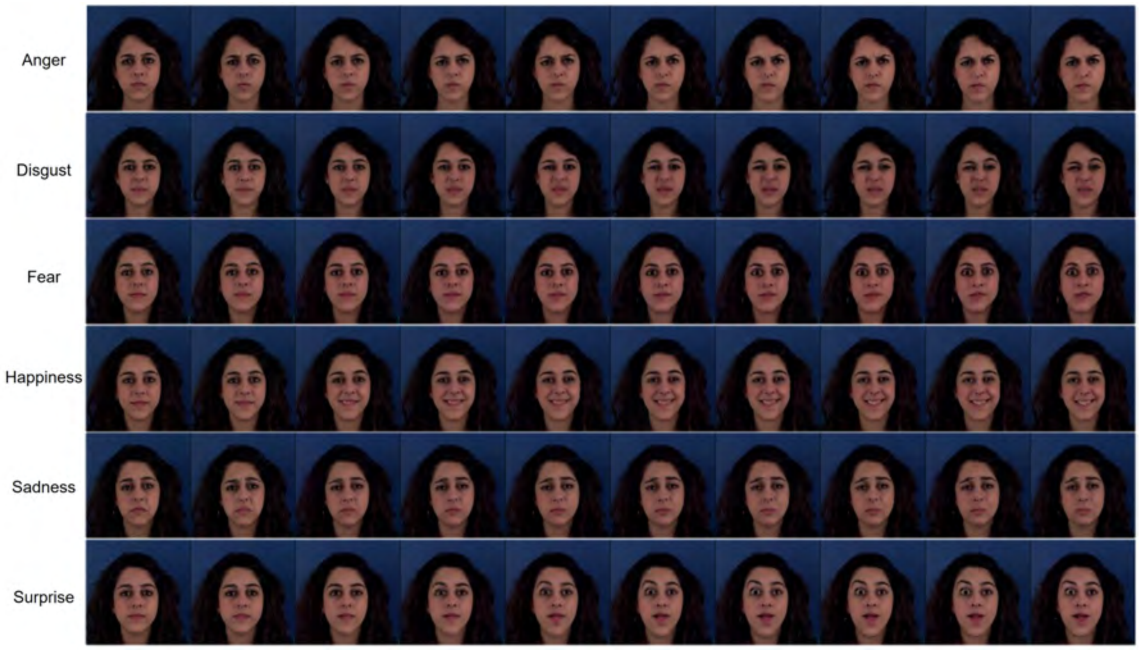


Figure 6.4: Single expression sequence generation frame-by-frame across 6 emotion classes (anger, disgust, fear, happiness, sadness, surprise). Our network is able to generate targeted emotion class based on the training data. The identity of the subject remained the same throughout the generation iterations. Note: First and second images are the input frames from MUG dataset, all the subsequent frames are generated.

Image quality assessment is performed and compared across all generation methods. Image quality assessment are used to determine whether the quality of the images is similar between two images. SSIM for full reference-based analysis and NIQE for no-reference based analysis are used. Full reference analysis provides a full pixel-by-pixel or region-by-region comparison of the similarity of two images; while no-reference analysis assess the image quality as a whole without the need of reference image. This gives a better overview of the quality of the generation which contains both reference and no-reference analysis.

For machine learning based method, no-reference image quality assessment is more suitable as this metric can distinguish distorted images from pristine images [222]. No-reference assessment is also not bound by pixel- or region-based differences. Ideally, the generated images should be realistic and novel, no-reference assessment measure the natural scene statistics (NSS) of natural images and compare with the generated image without the use of fixed localised reference. It is assumed that NSS of natural images has statistical regularity and comparing the NSS between images can measure the realism of images.

6.2.3 Experiments and Results

Training The network architecture is shown in Figure 6.2. We use 5 consecutive frames (2 initial frames for initialisation and 3 subsequent frames to capture spatiotemporal information). One constant neutral frame is fed to the network in every iteration for identity preservation. Mean square error (MSE) is also computed for each subsequent frame and generated frame. Training using 3 subsequent frames is essential as the network compares the generated data in 3 consecutive sequences which captures the temporal information.

All our network are trained without any labels. Instead, we feed in single category of facial expression (for Single Expression Sequence Generation) and all six basic expression classes (for Spontaneous Multi-Expression Sequence Generation).

Computational Cost

Our model is trained on NVIDIA RTX 3090. The training time for single expression is about 2 days. For mixed expression, it takes around 5 days. The size of this model is less than 100 MB (encoder is approximately 10 MB while decoder is around 80MB). The computational cost of the model is calculated using floating point operations (FLOPs). Our model has a complexity of 18.01 GFLOPs. Our model complexity are close to image based model which sits between Inceptionv3 [191] (11 GFLOPs) and ResNet [159] (21 GFLOPs). As a comparison, a video based model, Vision Video Transformer (ViViT) [223], has a complexity of over 4000 GFLOPs.

Results

Single Expression Sequence Generation Our model is able to generate realistic sequence (as shown in Figure 6.4) from a static frame and complete the expression sequence that involves onset, apex and offset phases (as shown in Figure 6.5). The identity of the participant is preserved throughout the generation. Our model is also able to generate across dataset with faces of different gender and ethnicity as shown in Figure 6.6. Each generated sequence are temporally correlated and when the expression reaches its apex, it is able to return back to the offset phase (neutral).

Spontaneous Multi-Expression Sequence Generation This network is trained with all six basic expression classes. In Figure 6.7, our model is able to generate multiple emotion classes of facial expression with two initial frames ("Fear" class). This shows the ability of our model

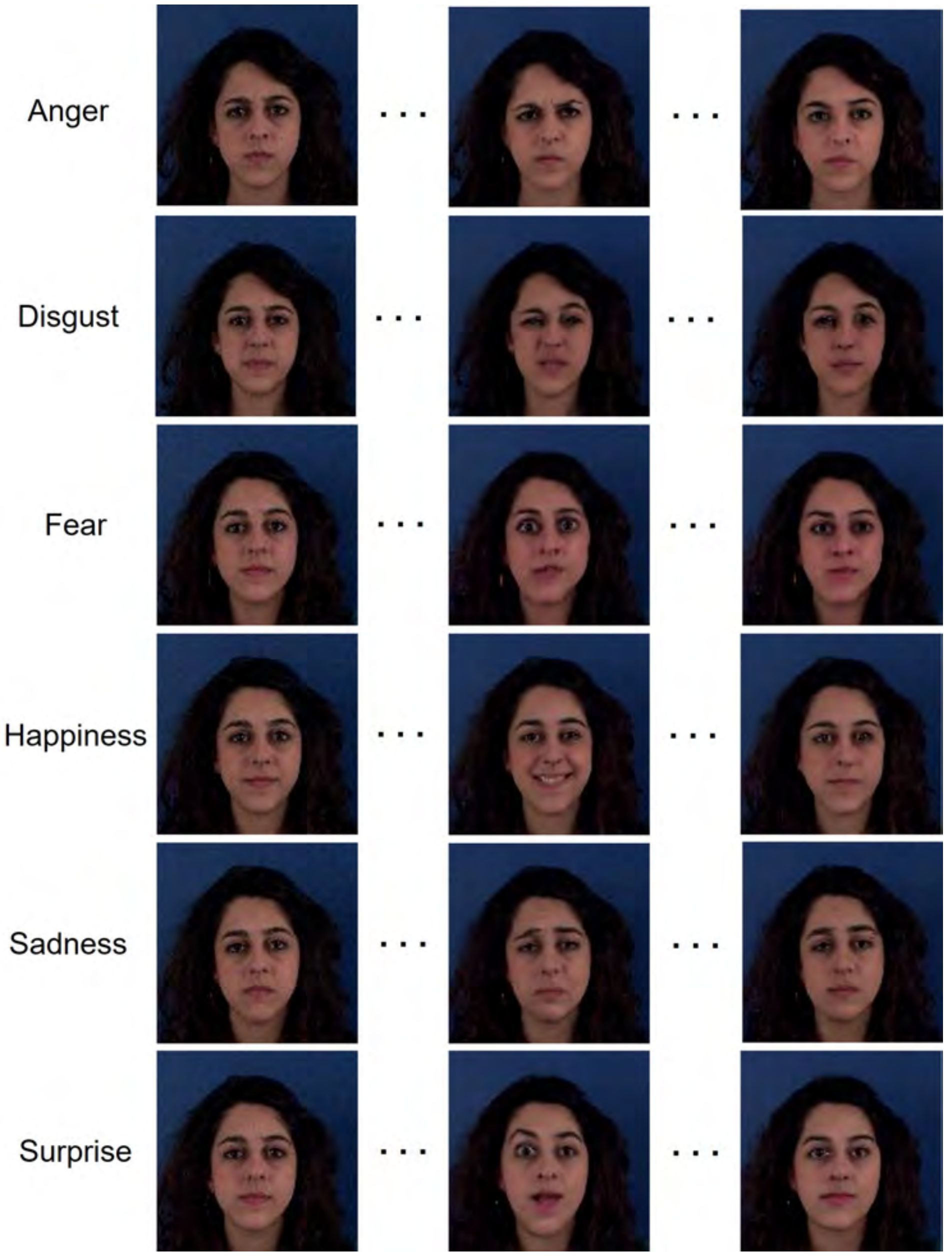


Figure 6.5: Single expression sequence generation of different emotional phase (onset, apex, offset) across 6 emotion classes (anger, disgust, fear happiness, sadness, surprise). Our method is capable of completing the entire sequence of facial expressions. Note: First image (onset frame) is the input frames from MUG dataset.

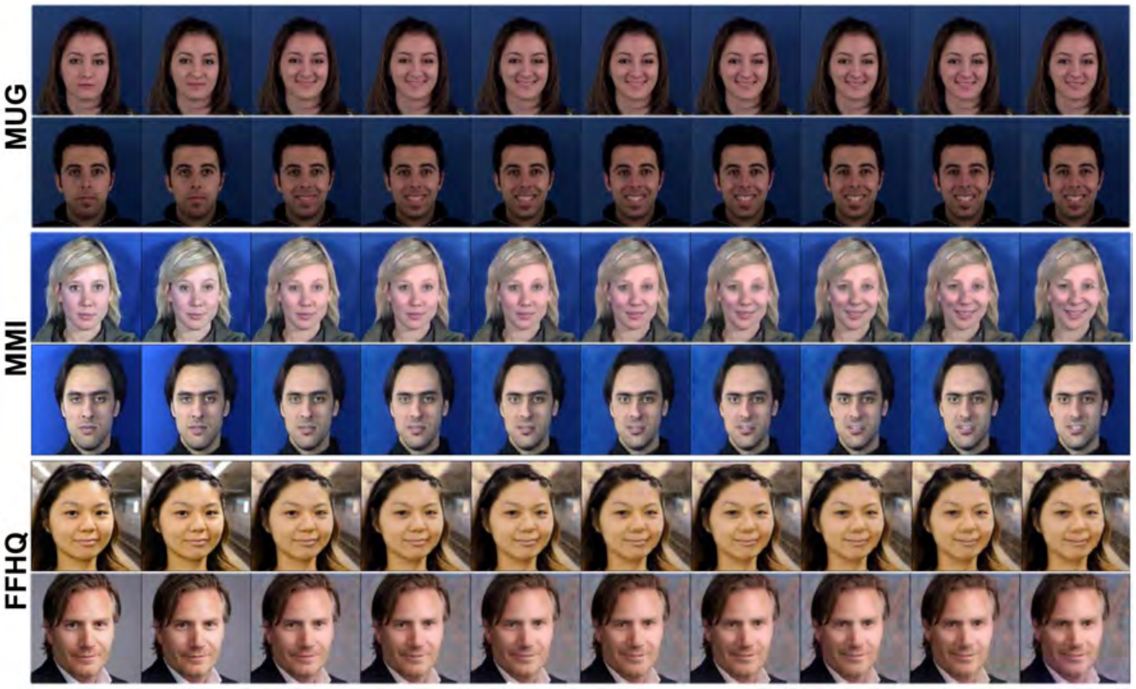


Figure 6.6: Comparison of single expression generation across datasets. Our method is able to generate facial expression across datasets (MUG, MMI and FFHQ dataset) with subjects of different gender and ethnicities. Due to FFHQ is an image dataset, all input frames are from a single neutral face. This demonstrates the ability of our model in generating facial expression sequences on unseen neutral faces.

to generate realistic sequence across different expression. This demonstrates that our network understanding on the realism of facial expression based on the training data.

Comparison with Other Methods

AU Comparison We compare the AU extracted by OpenFace for real and generated sequences. In Figure 6.8, the results show our method resembles real facial expression when compared to GANimation. The AU extracted to train this method is based on OpenFace. However, facial

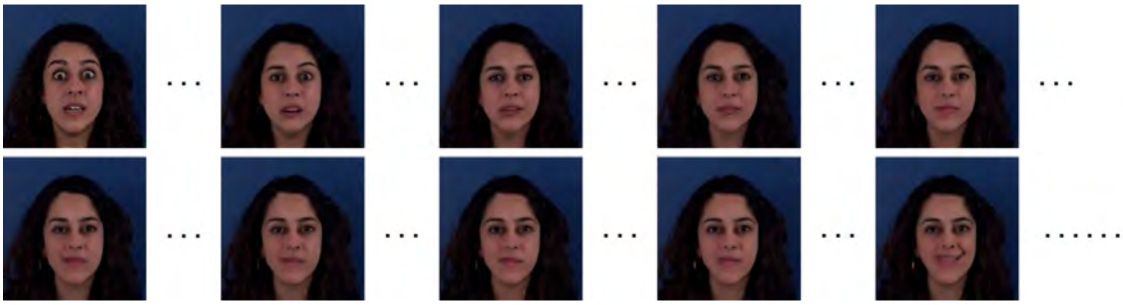


Figure 6.7: Spontaneous multi-expression sequence generation. Initialised using only 2 frames (fear expression). This version is trained using all six expressions. The generated sequences consisted of fear, followed by slight mouth movement (left side) and a broad grin.

expression is often non-linear. Hence, it lacks spontaneity, which is a common issue in guided facial expression generation. Our approach bypasses the need of labels and the use of linear variable. The results are more realistic as the network learns directly from the raw data and potential biases from another model (for AU extraction) can be avoided.

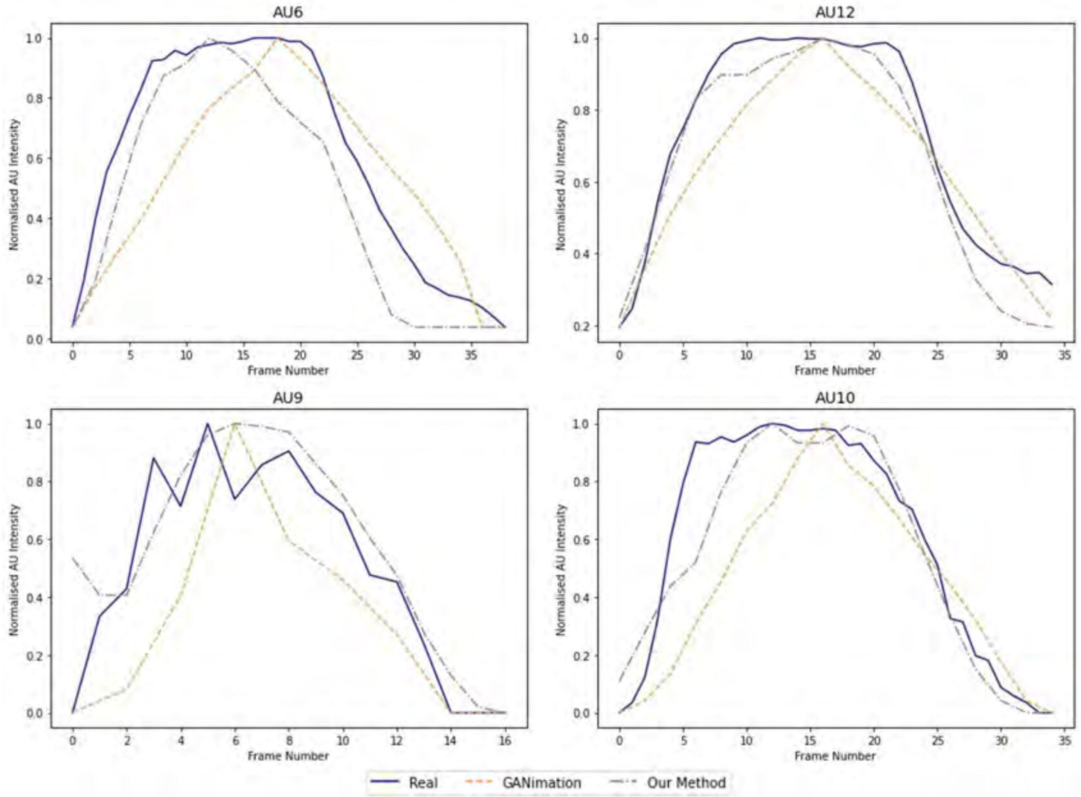


Figure 6.8: AUs comparison of real and generated sequences measured using OpenFace. Our method closely resembles AU of real facial expression with non-linear onsets and offsets. However, GANimation, a linear interpolation based facial expression generation method, shows constant AU intensity increment or decrement. GANimation generates expression on a per image basis and is fully guided by intensity input which differs from real sequence as shown. Note: AU6 and AU12 are from "Happiness" while AU9 and AU10 are from "Disgust".

Image Quality Assessment The results are shown in Figure 6.9. Our method outperforms all other facial expression generation methods in Natural Image Quality Evaluator (NIQE) while exhibit similar performance in Structural-similarity index (SSIM) with GANimation. These image quality are performed over full video sequences. For reference-based assessment (i.e. SSIM), our model and GANimation has similar average performance. For no-reference based assessment (i.e. NIQE), our model performs the best. This shows the sequences generated by our model has the highest image quality which is spontaneous and novel to the input frame (as this metric is not based on reference position).

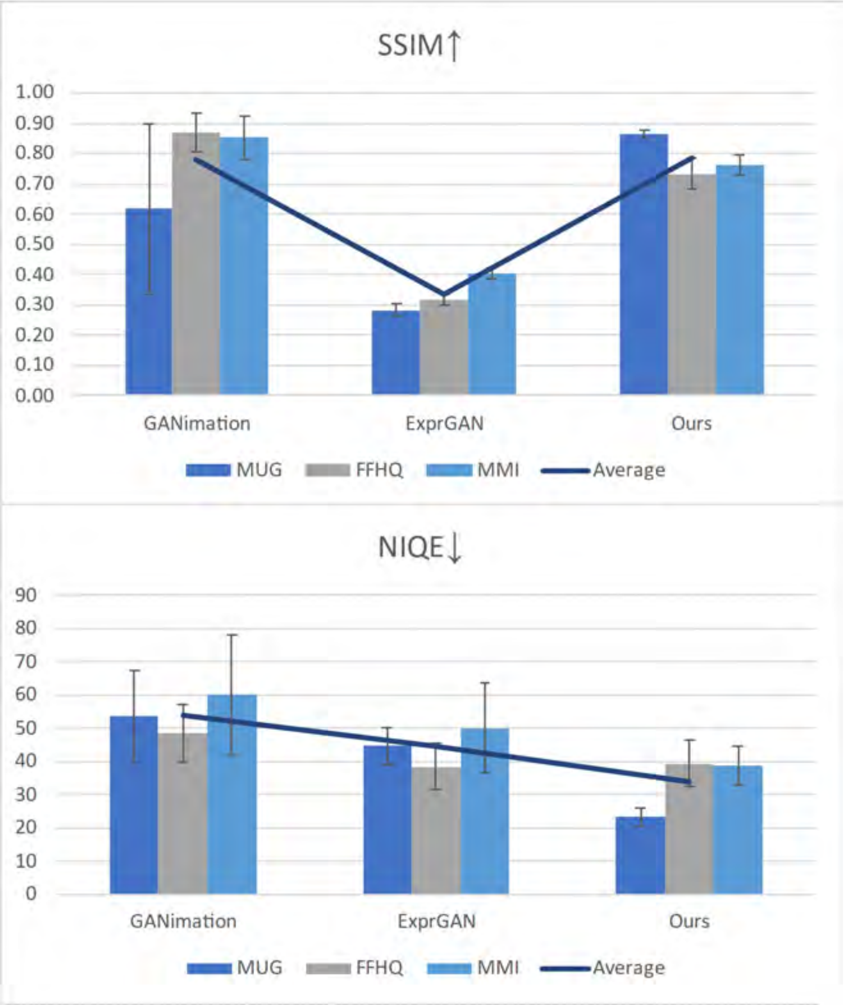


Figure 6.9: Image Quality Assessment using MUG, FFHQ and MMI as input. (Top) Comparison of SSIM, the higher the better. Our model has similar average performance across 3 datasets with GANimation. (Bottom) Comparison of NIQE, the lower the better. Our model has the best average performance across 3 datasets.

Comparison of Six Basic Emotion We compare the appearance of the apex of the facial expression with ExprGAN and GANimation. Both of these method requires facial alignment and face crop that may remove certain facial part (especially the chin). They also uses linear variable to augment the facial expression intensity. Although ExprGAN claims their model is able to preserve the identity of the input face, but based on the model uploaded by the author, the results is contradictory. Whilst GANimation is able to preserve the identify, it generates "Surprise" as "Anger". We show that we are able to generate facial expressions without the need of facial alignment or face crop. Our generated results are also novel as our approach allows the network to decide the facial expression that represent each emotion which is a stark contrast to the common approach that tune a linear variable to generate facial movements.

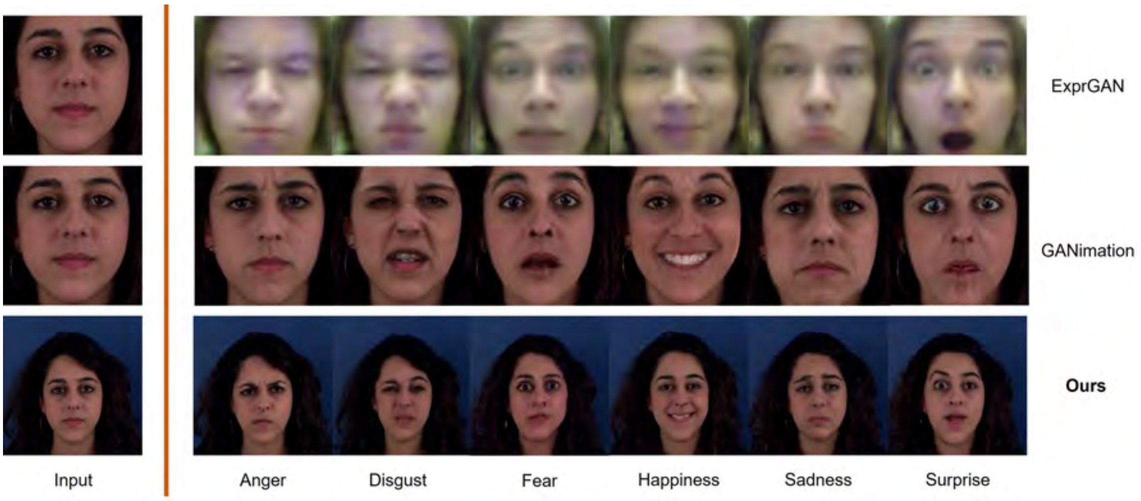


Figure 6.10: Comparison of single facial expression generation on MUG dataset. Six basic emotion are tested. ExprGAN completely changes the identity of the participants. Note that GANimation failed to generate "Surprise" class. ExprGAN and GANimation require facial alignment and uses a linear variable to tune the facial expression intensity. Our method is able to generate realistic expression based on each emotion class without any labels or guidance.

6.2.4 Ablation Studies

Removing the constant frame The effect of generation without using a constant frame can be seen in Figure 6.11. The identity of the face changes slightly. The facial expression quality also reduces drastically as the generation goes on. This shows the constant frame is essential for retaining the complex facial features.

Changing input order We investigate the effect of input frames sequence by swapping the order of frame 1 and frame 2. From Figure 6.12, we observe that the output results are not the same. Hence, we conclude that input sequence in our model matters. Our model has no issue with completing and regenerating more facial expression sequence in both cases.

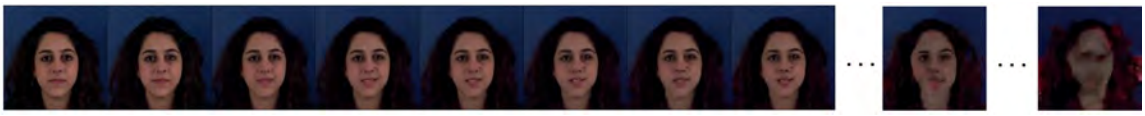


Figure 6.11: Ablation study: remove constant neutral frame. By removing the constant frame, the face identity changes and image become unrecognisable in latter stages of generation.

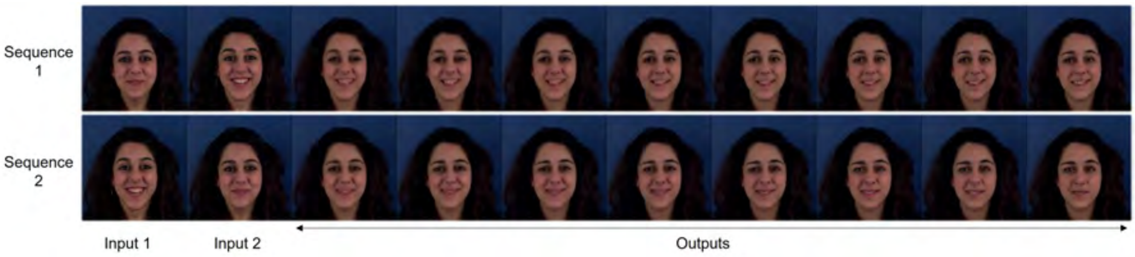


Figure 6.12: Ablation study: Changing input order. The effect of input order compared by swapping input 1 and input 2. The input order of the model matters as the generations are different.

6.2.5 Conclusion

We presented a novel latent representation method for facial expression sequences (of different emotion class) generation. Our proposed method is better than the existing methods as it is end-to-end, i.e., eliminate the need of pre-processing stage. Once the model is trained, it does not rely on labels or guidance from other facial sequences during generation. Our model is able to generate and complete the entire sequence of facial expression with only two input frames and a neutral frame. We demonstrated that it works even if we use a single neutral frame as the inputs. We also proved our generated sequence resembles the real facial expression using AUs comparison. Our model works with multiple faces across different gender and ethnicity. Our model is also computationally inexpensive when compared to video-based approach.

6.3 ME Generation

In this section, we discussed our spontaneous ME generation process using the same network architecture in Section 6.2. Instead of using MaE with distinct differences between each frame, we attempted to generate ME with very subtle frame differences. We tried two different training procedures in organising the input training set.

In the first attempt, we generate ME by applying the same architecture trained on long video datasets containing ME (SAMM-LV and CAS(ME)^2). We did not perform any input data selection and trained our network by looping through all the available video sequences. Based on the move-to-neutral ratio in Section 4.2.1 of the analysis of these datasets, the majority of the video sequences consist of neutral frames that contain no meaningful facial movements. As most of the training sets are static frames, this network (trained with self-supervised learning) tends to generate static frames as a result.

In the second attempt, we train our network by selecting training data from a short video ME dataset, CASME II. This is done by sampling the training data around the apex frame. This ensures that the sequences around the apex frame which contain most of the information of ME are included in every training step. This sampling is done by selecting between -5 to +5 frames around the apex frame and followed with choosing the training frame sequence with ±8 frames around the selected frame to form a set of 5 image sequences. The position of the selected frame (sampled around the apex frame) and the other randomly selected frames are fully random. The selection range of the input frames is chosen arbitrarily. The main idea is to select training frames around the apex frames that carries the most information on the ME related movements. The selection process is visualised in Figure 6.13. This ensures the model is trained with the whole sequence of the expression which in theory will enable the network to generate the entire sequence.

6.3.1 Results and Discussion

The results are shown in Figure 6.14. The model is able to produce spontaneous ME sequences without any labels. The subtleness of generated ME is compared with generated MaE (from "Surprise" facial expression from Figure 6.4) in Figures 6.15 and 6.16. The onset and apex frame of each facial expression respectively are computed for both analyses. The optical flow value is obtained by taking the third channel of the HSV colour map of the optical flow. Optical flow value describes the magnitude of the optical flow and in our application, it indicates the intensity of the facial movement. Absolute frame difference is the absolute value of pixel-by-pixel subtraction between two images. This analysis shows the difference between two images on a per-pixel basis. Standardisation and smoothing were used in our attempt to provide a more explainable visualisation of the output of this process. The standardisation involves subtracting

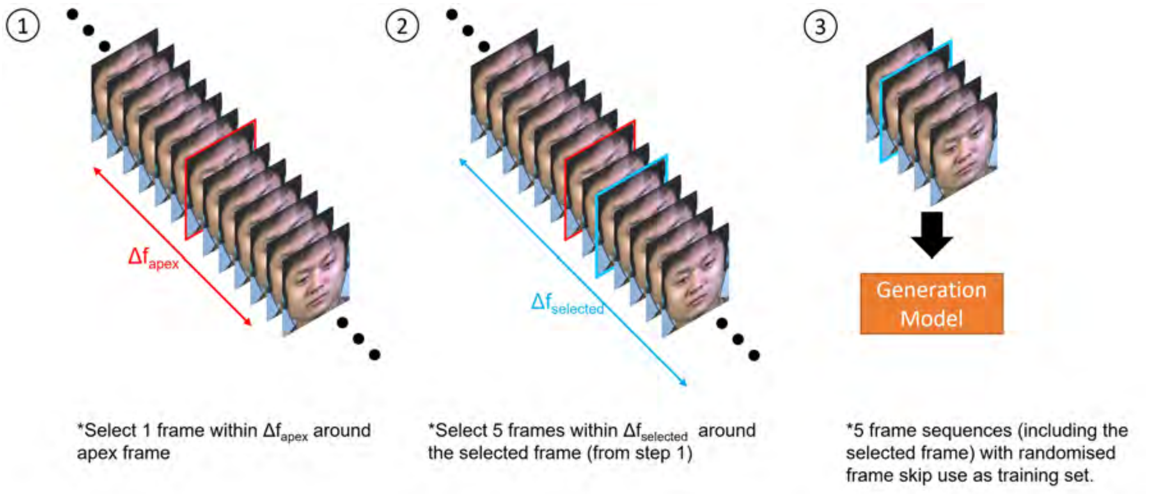

1
Δf_{apex}
*Select 1 frame within Δf_{apex} around apex frame
2
$\Delta f_{selected}$
*Select 5 frames within $\Delta f_{selected}$ around the selected frame (from step 1)
3
Generation Model
*5 frame sequences (including the selected frame) with randomised frame skip use as training set.

Figure 6.13: Frame selection around the apex frame for the training set for ME generation. Step 1: selects a random frame with a range of ± 5 frames around the apex frame (in red). Step 2: selects 5 random frames within the range of ± 8 frames around the selected frame (must include the selected frames). Step 3: 5 frame sequences with randomised frame skip are fed into the network. This training set selection method ensures every part of ME sequences is fed to the network.

the mean and dividing by the standard deviation of the pixel intensity over the entire image.

Comparing the optical flow value and absolute frame difference of both expressions, it demonstrates the subtleness of generated ME with respect to generated MaE. The absolute frame differences can only be general guidance, as the output in the generative model might not be identical due to the spontaneity of the model. The majority part of the absolute difference especially the background is black in colour which indicates there are no differences between frames and hence no movements. The subtleness of the generated ME is challenging to be detected even by experienced FACS coders.

We also provide a few other examples of ME generated on different parts of the face in Figure 6.17. This shows our model is capable of generating different ME sequences. Each movement is distinct from one another too. We demonstrated the facial movements using optical flow as the movements are visually similar to one another due to the subtleness of movement generated. The hotspots near the edge of the upper part of the image are due to the headphones worn by the participants of CASME II.

Besides performing ME generation on CASME II dataset, we attempted on another image dataset. A few static faces from AffectNet dataset also show great results over different movements being generated which were shown in Figure 6.18. Optical flow and absolute frame

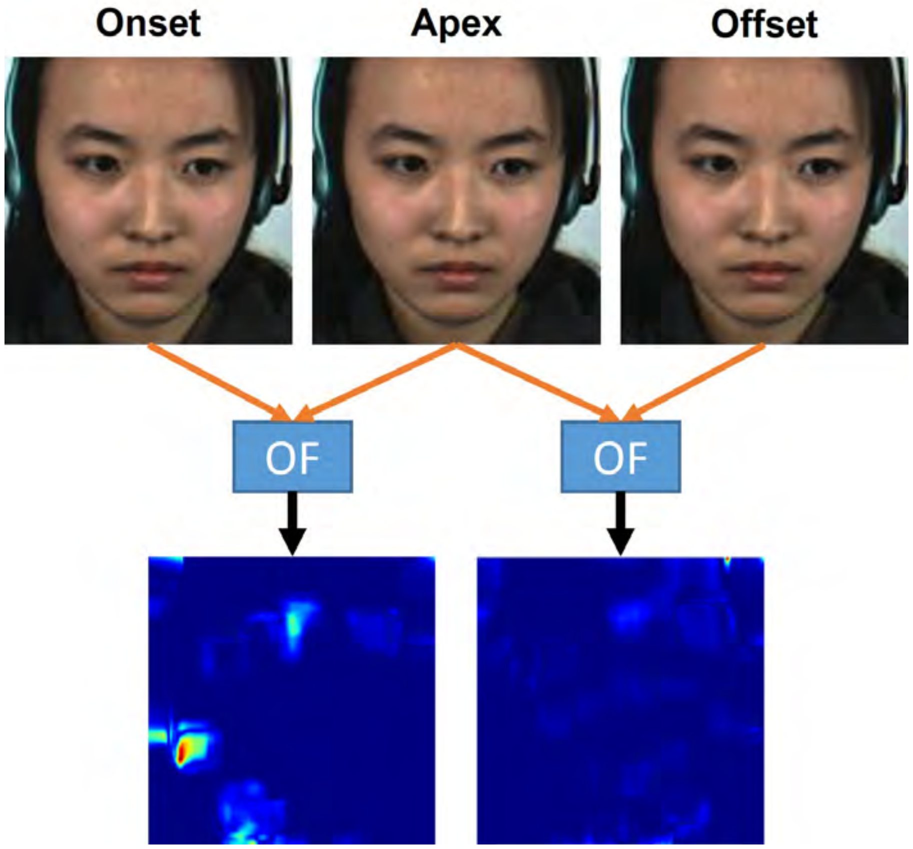


Figure 6.14: Spontaneous generation of ME generation train and validated on CASME II dataset. The onset, apex and offset frames containing ME are selected. Optical flow value computed between the frame sequence shows subtle movements in the form of a heat map highlighting the right eye of the subject.

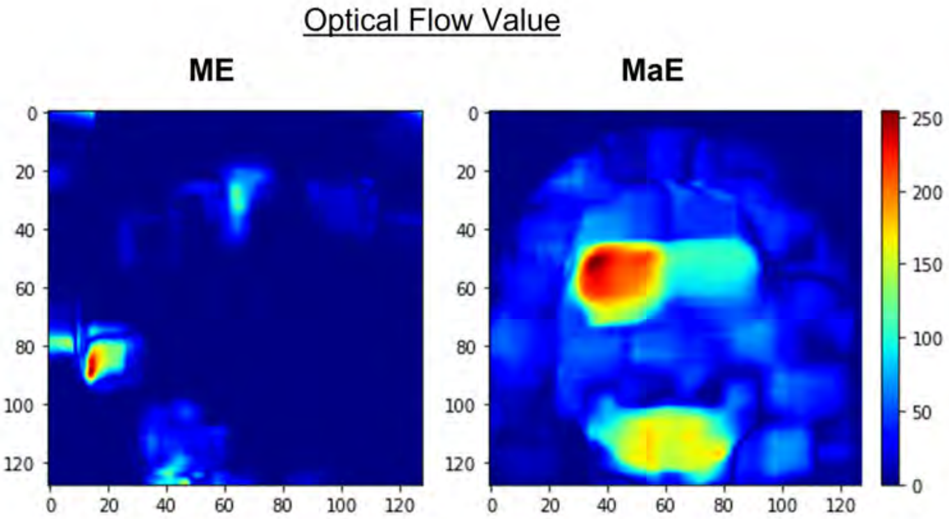


Figure 6.15: Optical Flow Value comparison between ME and MaE generated. Generated MaE is from "Surprise" facial expression from Figure 6.4. The onset and apex frame of each respective expression is used to compute the optical flow. The "Value" is the magnitude of the optical flow vector which describes the intensity of the movement. These images are normalised using min-max scaling of both optical flow values. Both selected facial expressions involve movements around the right eye.

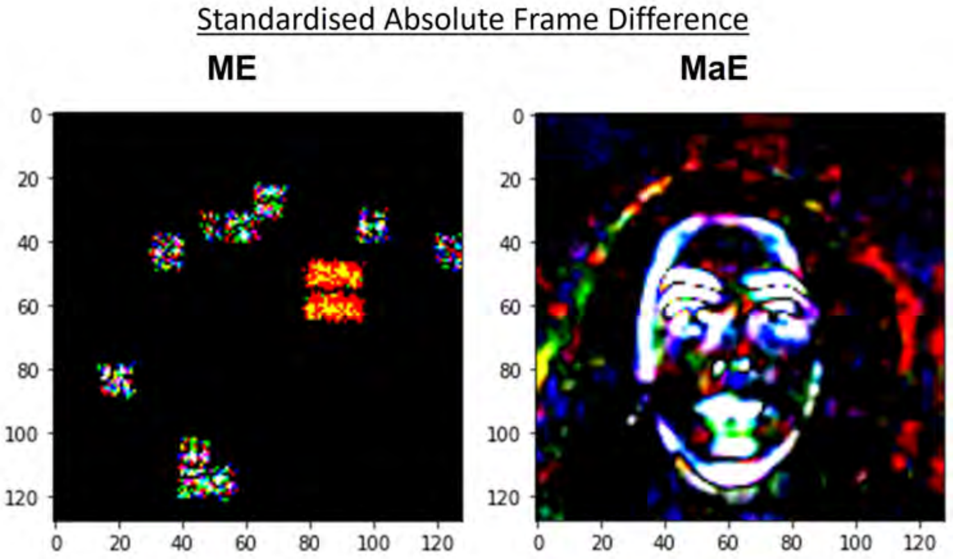


Figure 6.16: Standardised Absolute Frame difference comparison between onset and apex frame for both ME and MaE generated. Generated MaE is from "Surprise" facial expression from Figure 6.4. The onset and apex frame of each respective expression is used to compute the frame difference. These images are standardised (subtract mean and divide by standard deviation) and smoothed using Gaussian blur with a kernel size of 5. Both selected facial expressions involve movements around the right eye.

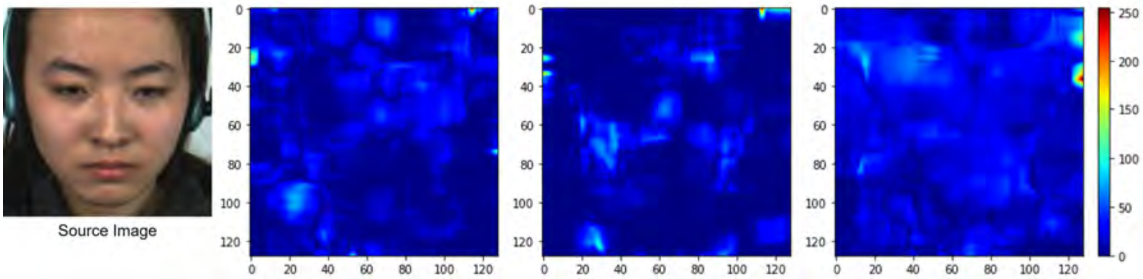


Figure 6.17: Optical Flow Value of ME generated. (Leftmost image) is the original input image. (2^{nd} image) involves movement of the right dimpler. (3^{rd} image) involves movement on the right cheek. (Rightmost image) involves right brow movements and left dimpler. Note: The hotspots on both sides of the top of the image are caused by the uneven brightness of the headphone worn by participants of CASME II dataset.

difference are visualised. Our approach also shows great generalisation across input faces of individuals of diverse ethnicity and gender. These motion analyses demonstrate that our model is capable of generating different MEs with very subtle intensity similar to the evaluation of CASME II subject. The hotspots in the optical flow value analysis is fewer compared to CASME II as the selected individual of AffectNet is not wearing any facial accessories (i.e. headphone).

This method is able to generate ME with subtle intensity. There is still room for improvement in the realism of this approach. The lack of datasets that contain ME is the main bottleneck for this approach. The subtleness of ME also poses a great challenge for determining the realism of the ME generated. There is still a lack of performance metrics to quantify the realism of ME other than manual human annotation or perception.

There are a few issues that we found in our method. First, our model will generate static frames (still images) without any motion. Next, some generated movements repeated in certain intervals forming a loop of movements as shown in Figure 6.19. The whole generation sequence remained static other than the facial movement that repeats indefinitely. The reason for this is still unclear.

6.3.2 Conclusion

We show that our sequence generation model can be adapted to generate ME by selecting the input frame sequence that contains MEs. As the majority of the frame sequences in ME long video dataset are neutral frames that carry no information on the targeted expression, ME short video dataset is used alongside input frame selection that centred around the apex frame. This selection process reduces the tendency for the network to generate image sequences that contains only static images.

6.4 Generalisation on Other Domains

The main limitation of our generative model is it is very dependent on the constant reference image. The effect of the absence of the constant reference frame can be seen in Figure 6.11. First, we investigate the generalisation of this model toward other sequential-based datasets. This furthers our understanding toward the mechanism of this model. Next, we attempt to remove the use of the constant input frame by introducing a new network architecture. This

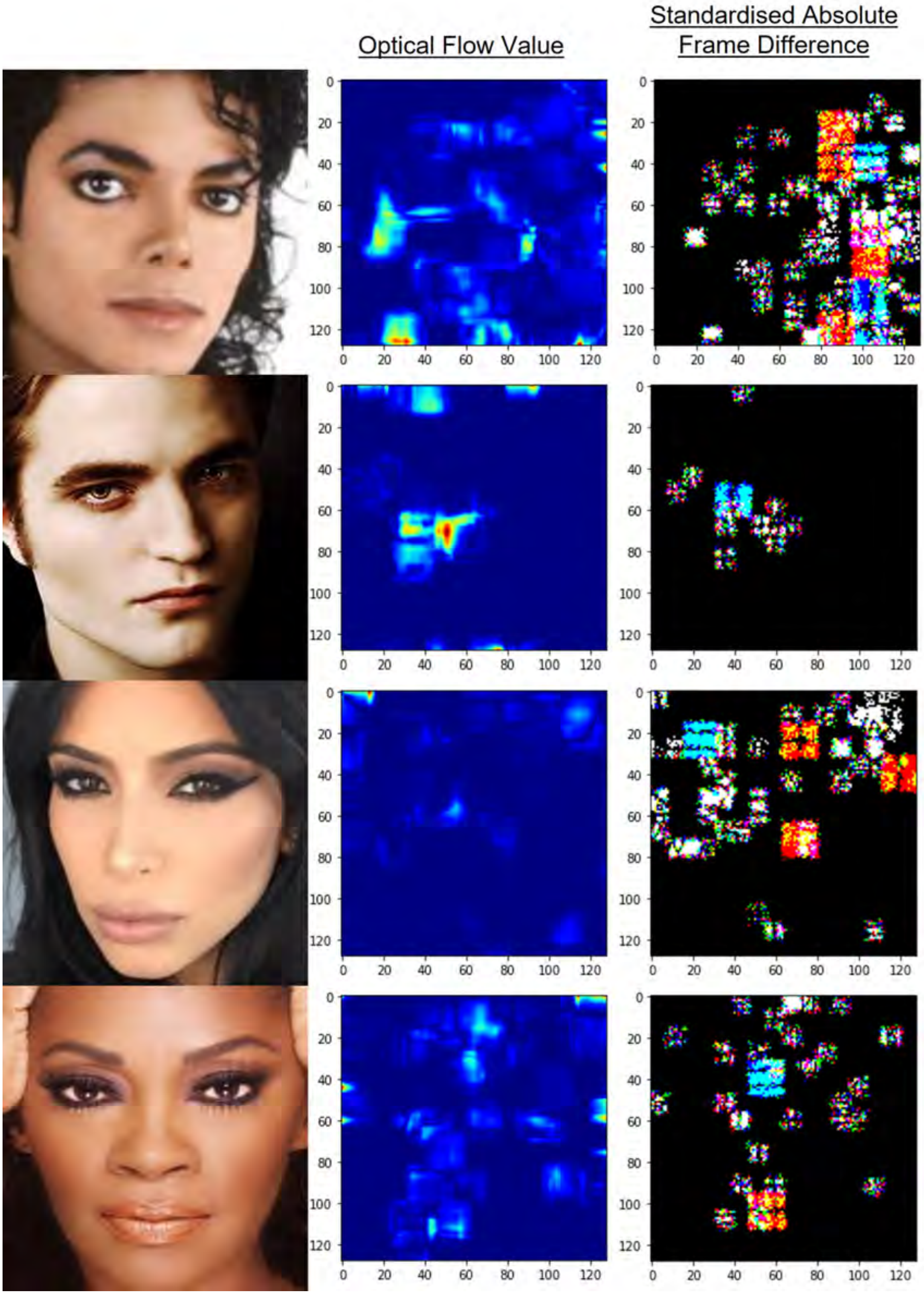


Figure 6.18: ME generation using AffectNet dataset. The Optical Flow Value and Absolute Frame Difference show different ME being generated with very subtle intensities.

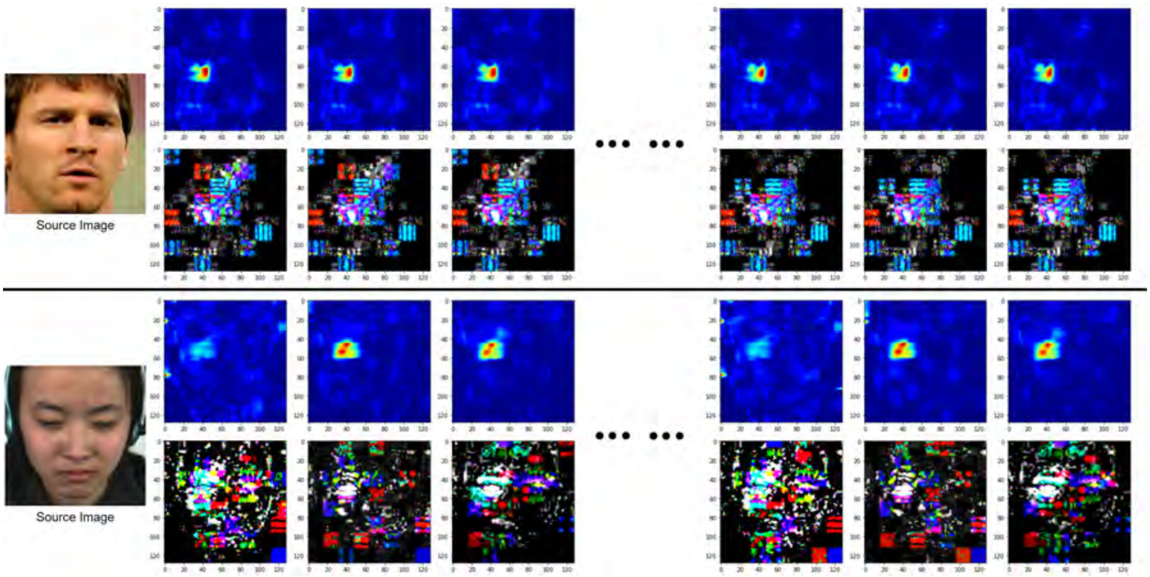


Figure 6.19: Examples of ME generation that generates repeated movements for the entire video.

network is consisted of a repeated network structure of the previous method by taking inputs of a certain sequence of images (instead of 2 used by the last approach).

The motivation behind this study is if the generative model can work on sequences without the need for a constant reference frame, it can potentially be applied in the prediction of real-world scenarios where there is no constant reference frame to be referred to.

Dataset

We use Moving GIFs and Moving MNIST to investigate our method further. The dataset information is stated below.

Moving GIFs (MGif) [224] dataset consisted of 1000 videos (in gif format) containing movements of different cartoon animals. This dataset is diverse in motion and visual appearance of the moving characters. Majority of the character movements are walking animations.

Moving MNIST (V-MNIST) [225] contains 10,000 video sequences. Each video has resolution of 64×64 with a length of 20 frames. Two digits move independently around the frame with no collision and they overlap when in contact with each other.

Generation using only Latent Representation

This model used has a similar structure to Figure 6.2. However, the model was downscaled to using only 3 latent representations and the image input dimension is 64×64.

This model can generate the next frame and complete a movement sequence using MGif dataset as shown in Figure 6.20. MGif dataset is very challenging as this dataset consists of characters with vastly different visual representations and movements. The visual quality of recursive generation is lower compared to faces as a result of the diversity of appearance and movements of this dataset. We found that the generated motion is more realistic if the limbs of the characters are distinctly visible to each other. The majority of the movements in this dataset are near the limb regions. Our model recognises limbs easier when they are distinct from each other and performs movement on each limb which makes the movement more realistic. For characters with limbs that are indistinguishable from each other, the motion generated as less realistic (as shown in Figure 6.21). One of the possible reasons is the original walking animation of these characters is very different from other characters of the training set. Hence, the model did not predict the next motion of the generation accurately.

Upon further inspection, we found that our model is performing pixel-based augmentation. When the limbs are visible, this model has learnt to change the limbs to simulate movements. However, due to the constant reference frame, the generated movements are localised to a certain region of the image. These localisation of movements generated is the nature of this dataset where the character is situated in the middle while performing movement animation.

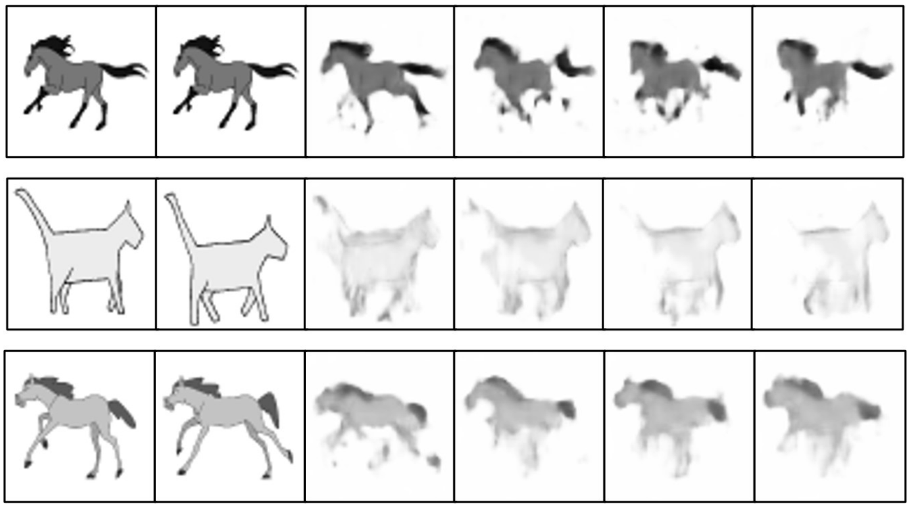


Figure 6.20: Successful attempt of sequence generation using mgif dataset.

We perform ablation studies by augmenting the input data using horizontal and vertical flips. The results are shown in Figure 6.22. Upon flipping the input images horizontally (left and right), the generated images remained realistic. The generation remains realistic despite

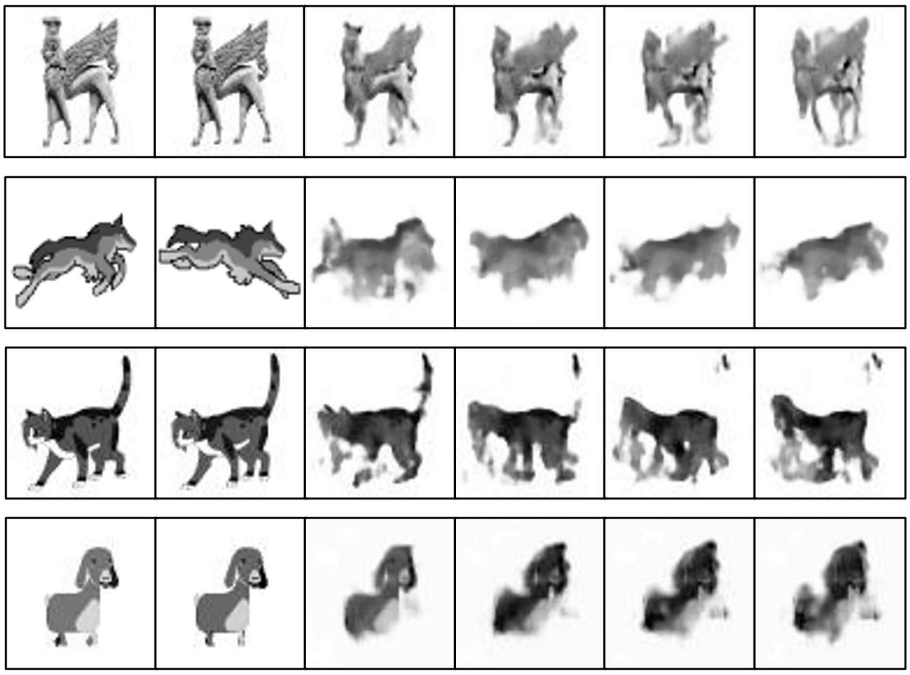


Figure 6.21: Unsuccessful attempt of sequence generation using mgif dataset.

the change in direction (left or right) of the character. However, flipping the input images vertically (upside down) makes the quality of the generated sequence becomes worse. The trained network identifies the localised top and bottom structure of the character. Hence, the generated sequence appears worse as it attempts to generate moving limbs at the bottom of the image (the top position of the character).

Cascaded Auto-Regressor without Constant Input

The previous generative model is positional sensitive, for example, if the face or characters are not placed in a similar position or slightly out of position as the training set, the generation will not be realistic. We also experimented without using a constant reference frame to address generation across different positions. This version of the network takes in multiple inputs from the image sequences using multiple structures inspired by our previous method in Figure 6.2. The generation is visualised as in Figure 6.23. V-MNIST dataset was used as the first attempt to generate sequence without non-localised reference. This dataset was chosen as it contains objects that consisted of simple features (moving digits). V-MNIST works on non-overlapping individual digits. Due to the “memoryless” nature of Markov chain, without any external reference, the network is unable to maintain or remember the structure of digits after overlapping. Instead, the identity of the digits will either change or break down completely.

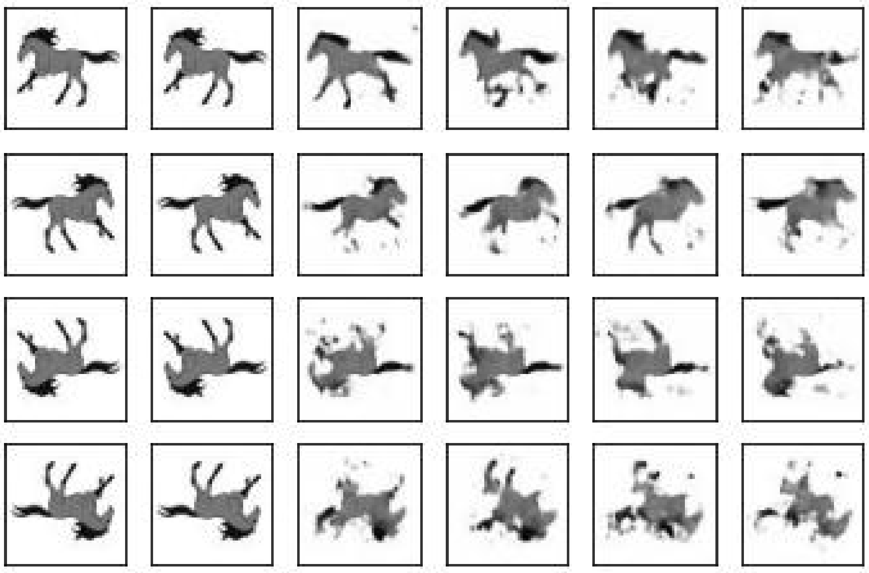


Figure 6.22: Sequence generation using mgif dataset with horizontal and vertical flip augmentation. The generation remains realistic with horizontal flips while vertical flips degrade the quality of image generated significantly.

The examples of the generation are shown in Figure 6.24.

With the success of V-MNIST generation, we attempted a more complex version of this identical network with a higher number of filters on MUG dataset. Nevertheless, the algorithm will attempt to generate a face from the training set and performs sequence generation based on the face as shown in Figure 6.25.

The generation shows that the model turns the input images into an existing face from the dataset and performs facial expression generation on the face. This sequence generation model uses MSE as loss function. MSE compares on a per-pixel basis which will try to match the next frame based on the previous observation. Our model generates the next sequence based on spatiotemporal information (spatial and temporal information) on current observation.

In our sequence generation task, the ideal scenario is the model converge and generate the next sequence using temporal information (the next facial expression motion sequence) while retaining spatial information (identity of the subject). Without any labels or supervision of the spatial or temporal information, the network learns these information via self-supervised learning. These information embedded in the latent vector is indistinguishable to the model. Telling our model to separate spatial and temporal information in the latent vectors (which is a mix of spatiotemporal information) is essentially an inverse problem (this problem is visualised

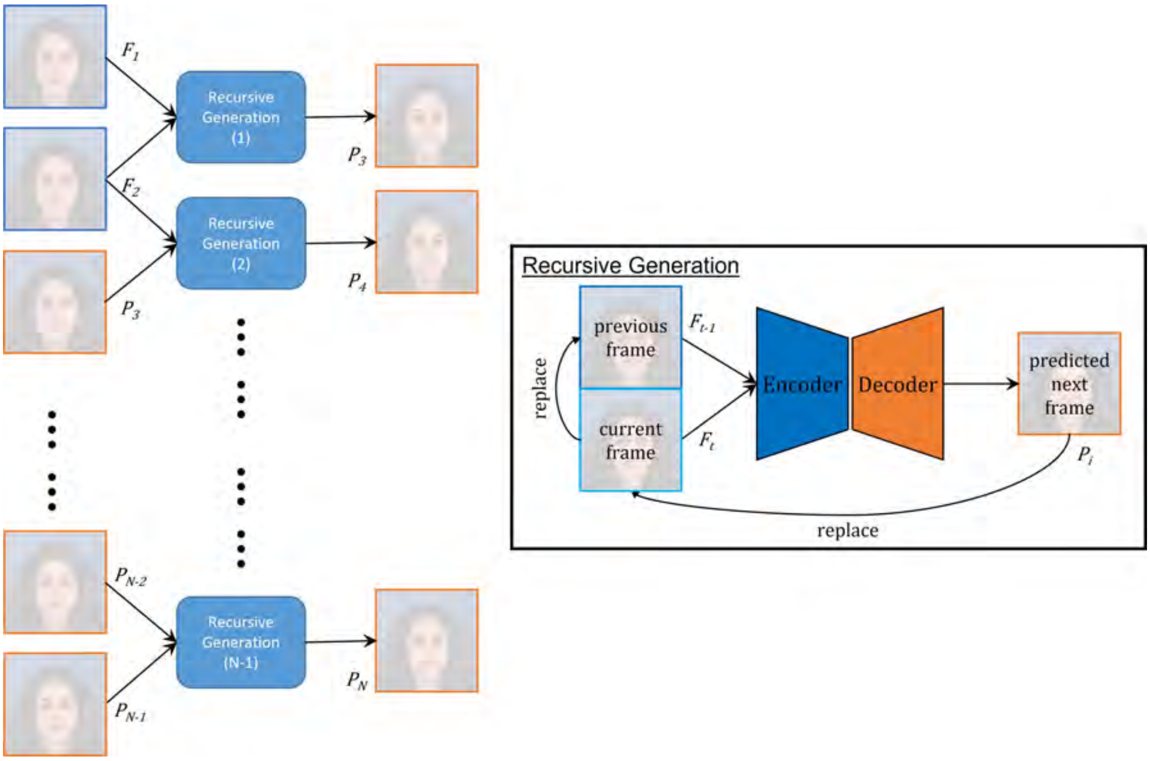


Figure 6.23: Cascaded Auto-Regressor without constant input. This network does not rely on any reference frame and generates recursively based on only current observations. This figure represents one generation loop. Note: F is the original frame, P is the generated frame, N is the number of frame sequences to generate per loop.

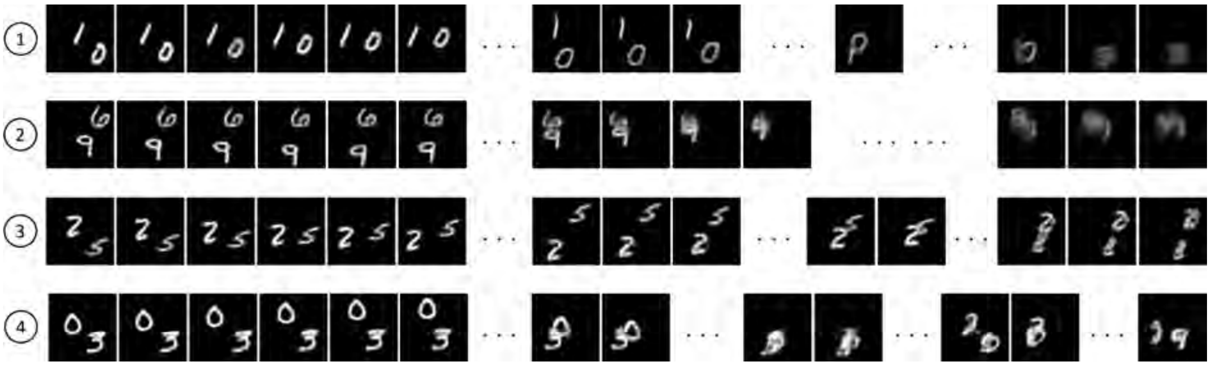


Figure 6.24: V-MNIST generation without any constant input. The sequence generation maintains the structure of digits when the digits do not overlap. The identity of the digits remains stable until overlapping happens. When overlapping occurs, the digits will either become unrecognisable (as in 1^{st} and 2^{nd} row) or turns into another digit (as in 3^{rd} and 4^{th} row).

in Figure 6.26). Our model was also not supervised explicitly to retain spatial information (identity of the subject). Hence, the Markov property of the sequence generation loop converges in either spatial, temporal or spatiotemporal dimension and does not retain spatial information which was intended but not explicitly instructed.

One other possible reason is the number of training subjects is only 51 (plus 1 participant as test set) which is relatively small. The model has a limited amount of facial expressions of different distinct individuals to learn from. This model might be overtrained on the identity of the subjects from the dataset and when it is evaluated on unseen data, the model will converge spatially to one of the training sets and perform facial expression generation.

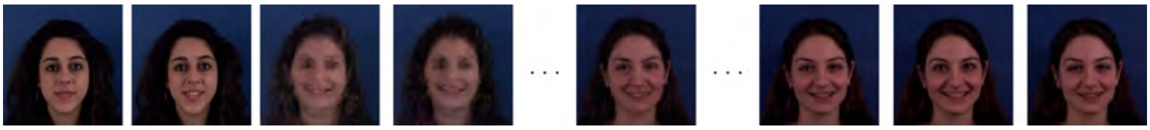


Figure 6.25: MUG generation without any fixed input. The identity of the subject changes and facial expression is generated on the face with the identity changed.

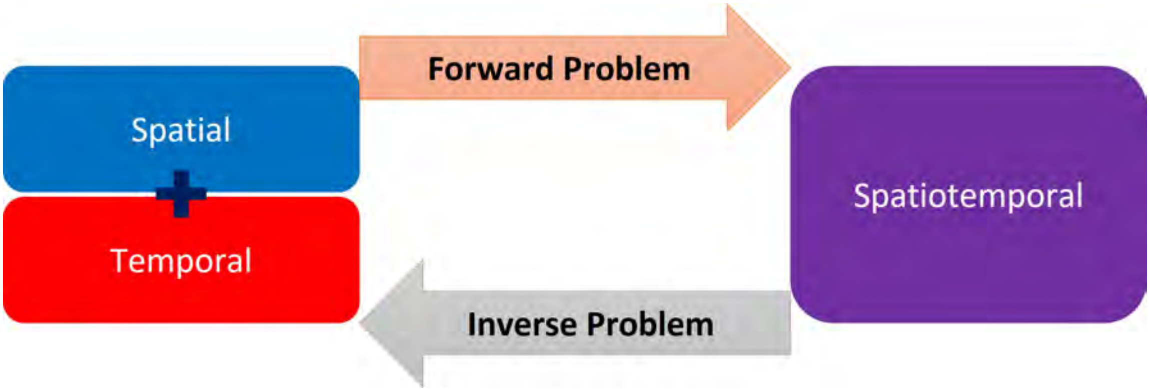


Figure 6.26: The forward and inverse problem in dealing with spatiotemporal information. Given a set value of spatial and temporal vectors, we can find the spatiotemporal vector effortlessly. However, if given a spatiotemporal vector and asked to separate it into the original spatial and temporal vector, the solution is not that simple. Our model was given spatiotemporal information (2 image sequences) and without any supervision was told to retain the subject identity (keeping spatial information about constant) while changing the subject movement (performing facial expression movement). This is essentially a complex inverse problem.

6.5 Summary

We introduced a facial expression sequence generation method capable of generating without using any labels. Before performing the full facial expression generation task, a simple naive approach is attempted to explore the recursive generation ability of the model. This naive model then becomes the backbone architecture for the recursive generation loop. Based on the latent representation extracted, our model is capable of completing a facial expression sequence using only three input frames (two image sequences and one fixed image). Our model is able to generate unique facial expression sequences when trained with single expressions. Our model can generate fully spontaneous expression sequences when trained with mixed facial expressions.

We attempted to generate ME using the same architecture by changing the training set to ME dataset. The training set (5 frame sequences) is randomly sampled around the apex frame in the same order. This ensures the network is always trained with ME sequences and the random selection ensures the entire ME sequence was covered. We show that our model can generate subtle ME. The generated MEs are visualised using optical flow values and absolute frame differences.

We further investigate our model by applying it to other sequence-based datasets. We also attempted to remove the constant reference frame of our model by taking in a larger sequence using the repeated structure of the original approach. The results show constant reference frame is essential in the current stage of our research. Our model convergence is based on the spatiotemporal information observed. Without the constant reference frame (constant spatial information), the sequence generation will take on spatial, temporal or spatiotemporal convergence for each iteration. Without any constant information, the identity of the subject changes as the model does not distinguish spatial or temporal information and hence, converged spatially.

Chapter 7

Facial Expression Recognition

To provide qualitative and quantitative evaluations on the quality of facial expressions generated by the generative model in Chapter 6, we use human assessment and computational methods. The ground truth of the generated videos is rated by a FACS coder. Two deep learning methods are used to predict the generated videos into positive and negative classes.

Facial expression recognition (FER) is a task to classify the facial expression into one of the emotion classes (such as happiness, sadness, surprise etc). This is important as it identifies and estimates the emotional state of a person using non-intrusive visual information. Commonly, it is treated as a classification task for machine learning based approach. In this chapter, we explored facial expression recognition as two different problems: classification and regression. The classification task will attempt to predict discrete labels while the regression task outputs continuous labels.

We evaluate both approaches on facial expression sequences with changing facial expression intensity. We propose these FER models to be potential methods as a preliminary analysis for fully automated facial expression sequence analysis system. This method will be a part of our pipeline towards a fully autonomous facial analysis toolkit.

7.1 Training Dataset

The training dataset used in this section consists of images labelled with facial expressions.

FER2013 (Facial Expression Recognition 2013 Dataset) which consisted of 28,709 training and 3,589 test examples of 48×48 pixels grayscale images is used for prototyping. There are 7 emotion categories (Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral) used in this dataset.

Extended Cohn-Kanade (CK+) dataset [226] consists of 593 sequences from 123 participants. The video sequences are recorded at 30 fps with a resolution of either 640×490 or 640×480 pixels. Only 327 videos are labelled with one of the seven emotion classes: happy, sad, anger, disgust, fear, surprise and contempt. This video sequence starts with the onset frame and ends with the apex frame where the intensity of expression is at its peak. In this experiment, we only take the apex frame of the dataset and used this dataset as static images.

The Karolinska Directed Emotional Faces (KDEF) [227] is a facial expression image dataset. This dataset contains 490 images with a resolution of 562×762 of 70 individuals (35 females and 35 males). It is annotated on seven emotion classes: happy, sad, anger, disgust, fear, surprise and neutral. The videos are captured at 5 different angles, only frontal view was used for our experiment.

AffectNet [228] is a large-scale facial expression dataset that consisted of more than 1 million collected by searching 1250 emotion-related keywords in six different languages. About 0.4 million images are manually annotated for eight emotion classes: happy, sad, anger, disgust, fear, surprise, contempt and neutral. The average image resolution is 425×425 pixels.

The training dataset is a composite dataset consisting of FER, CK+, KDEF, FFHQ and AffectNet. FER, CK+ and KDEF consist of grayscaled facial expression images. FFHQ and AffectNet are consist of facial images with colour, however, the emotion class label on FFHQ was annotated using a facial expression recognition algorithm as the original dataset was not annotated with emotion with respect to each image (Source: `https://github.com/DCGM/ffhq-features-dataset`). For FER, CK+ and KDEF, the image resolution of these datasets are standardised at 224×224 pixels before resizing.

Modified Dataset Labels

As all datasets are labelled with different classes, we modified and standardised all the training labels to three simple emotion classes (negative, neutral and positive). "Negative" class consists of anger, disgust, fear, sadness and contempt; "Neutral" class consists of neutral and surprise; while "Positive" class consists of only happiness. The motivation for making this classification is to quantify the emotion class of the expression sequence generated.

The visual representation of the dataset of each class distribution is shown in Figure 7.1. We collect as many annotated facial expression image data for this task. As AffectNet is the largest available facial expression dataset, this dataset becomes majority of our training data especially in "Negative" class where the other dataset has negligible amount of samples.

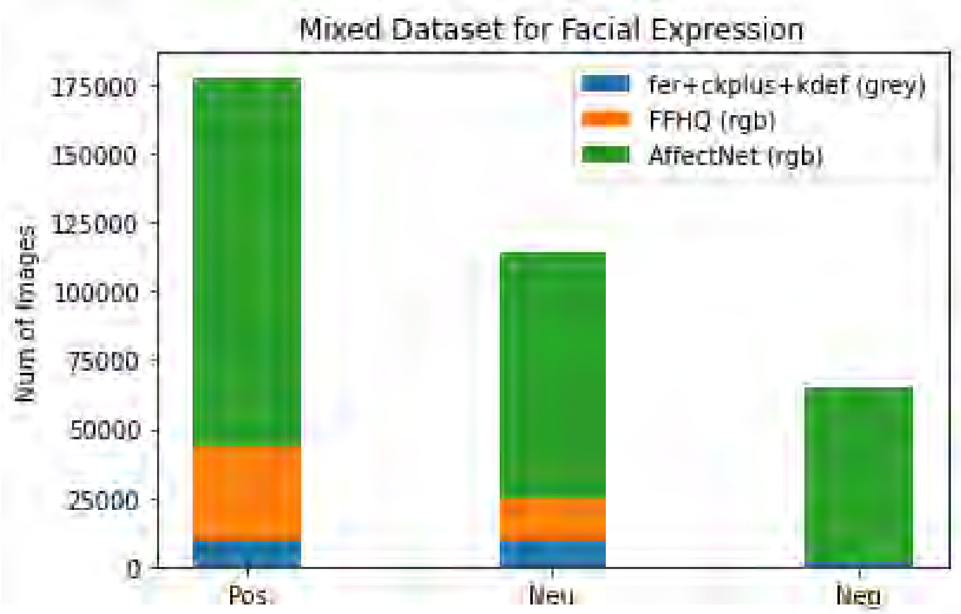


Figure 7.1: Facial expression recognition training set which is a composite dataset consisting of FER, CK+, KDEF, FFHQ and AffectNet. AffectNet has the highest amount of dataset count followed by FFHQ. FFHQ is annotated using a facial expression recognition algorithm. The emotion classes are: "Negative" class (Neg) consists of anger, disgust, fear, sadness and contempt; "Neutral" class (Neu) is consisted of neutral and surprise; while "Positive" class (Pos) is consisted of only happiness.

7.2 Methods

Inspired by the facial expression analysis by OpenFace (where OpenFace analyses AU presence as a classification task and AU intensity as a regression task), we attempted two approaches on the facial expression recognition task whereby we treat this task as either a classification or regression task. The reason of attempting both approach is to investigate which method performs better and more suitable in recognising facial expression sequences. This is done by evaluating on original MUG dataset and the generated sequence generated by our spontaneous generation method from Section 6.2.

For both attempts, the input and hidden layers are kept constant. For classification, the final activation of the output layer is softmax and the loss function is categorical cross-entropy. The labels used are "2" for "Positive", "1" for "Neutral" and "0" for "Negative". For regression, the final activation function is linear and the loss function is mean squared error. The labels used are 0.0 for "Negative", 0.5 for "Neutral" and 1.0 for "Positive".

The model used in these attempts is a standard convolutional neural network downsampled using convolutional layers with a stride of 2. The difference between these two networks is the output layer. The overall network architecture is visualised in Figure 7.2.

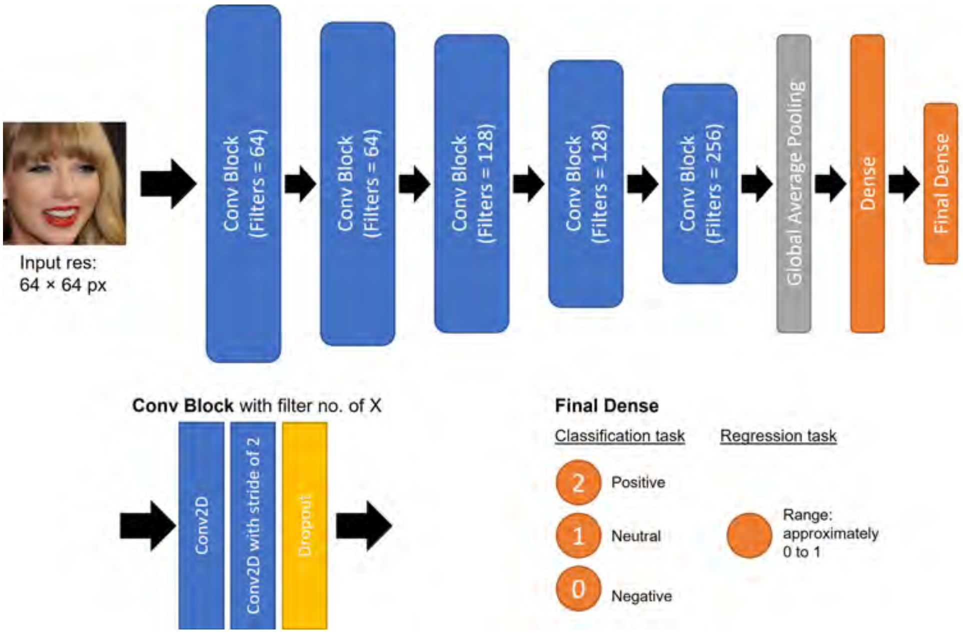

Input res:
64 × 64 px
Conv Block (Filters = 64)
Conv Block (Filters = 64)
Conv Block (Filters = 128)
Conv Block (Filters = 128)
Conv Block (Filters = 256)
Global Average Pooling
Dense
Final Dense
Conv Block with filter no. of X
Conv2D
Conv2D with stride of 2
Dropout
Final Dense
Classification task
Regression task
2 Positive
1 Neutral
0 Negative
Range: approximately 0 to 1

Figure 7.2: The network architecture is a vanilla CNN with downsampling using strided convolutional layers. The differences between classification and regression tasks are the final dense layers where classification has a filter number of 3 for each class while regression has a filter number of 1 which has a range of approximately between 0 to 1.

Regularisation techniques are used in this attempt. Random data augmentation that randomly modifies the brightness (with a maximum range of 0.1), contrast (with lower bound 0.2 and upper bound 0.4) and horizontal flip (with 50 % probability) are used. Weighted loss function was used to address the class imbalance of the training set. In classification task, minority classes ("Neutral" and "Negative" classes based on Figure 7.1) are weighted with a factor of 2 and 3 with respect to the majority class ("Positive").

This evaluation identifies facial expressions and attempts to sort them into three classes using either discrete classes (classification) or continuous classes (regression). We trained on static facial expression images and investigate the implication on recognising facial expression sequence.

7.3 Results and Discussions

Both FER **classification** and **regression** models are evaluated on the real facial expression sequences and generated image sequences. The comparison between both tasks is done by taking the raw output of the model without any post-processing.

Evaluation using Real Facial Expression Sequences

We evaluated the facial expression recognition algorithm on original facial expression sequences from MUG dataset that were used in Chapter 6. Please note that the length of video sequences are not identical. The output of the model is shown in Figure 7.3.

Quantitative Measurement

We evaluate both FER methods on 20 video sequences (10 from MUG dataset and 10 from MMI dataset). Our FER algorithms classify the videos into three classes and compare the prediction with the original ground truth. The accuracy of the FER methods is tabulated in Table 7.1. Both our models identify "Positive" class perfectly. The overall accuracy is 0.80 which shows our models are capable of identifying different classes of facial expression sequences despite training on facial expression images.

Table 7.1: Accuracy of FER Classification and Regression MUG and MMI video sequences. The performance is evaluated compared to the given ground truth. Both methods performs perfectly in identifying "Positive" emotion class.

	Negative	Positive	Overall
FER Classification	0.70	1.00	0.85
FER Regression	0.60	1.00	0.80

Qualitative Measurement

We further evaluate every emotion class and attempt to classify them into three simple emotion classes. The temporal dynamic (onset and offset) is investigated as we are interested in knowing how well the algorithm deals with changing facial expression intensity.

For FER **classification** task, there are no onset and offset of the facial movements as it is a classification with no intermediate values between each classes. Each classes are mutually exclusive to each other. All classes other than “disgust” were classified correctly. This shows that classification of three simplified emotion classes is less challenging compared to the regression task (which will be discussed in the next paragraph). However, this approach has the drawback of ignoring the intermediate transitions of facial movements of different emotions. The temporal dynamic is non-existent.

For FER **regression** task, the interval where “happiness” occurred was predicted accurately as “positive”. “Surprise” is predicted as “neutral”, “anger” and “disgust” is predicted as “negative” which matches the intended outcome. Both approaches show disagreement on “sadness” class where the regression task treats this expression as “neutral”. This model allows continuous transition between emotions which shows a more realistic facial expression analysis where it shows onset and offset of the facial expression.

Evaluation using Generated Facial Expression Sequences

We evaluated both FER algorithms on the image sequence generated using the “Facial Expression Generation using only Latent Representation” method in Section 6.2. This can potentially be a realism evaluation metric on our generated data from our method of previous chapter.

Quantitative Measurement

Similar to the evaluation on real facial expression sequences, we evaluate both FER methods on 20 generated video sequences. Although we generate both expressions equally (10 using the network with “Positive” weight and 10 with “Negative” weight), some of the generated videos appear visually appear similar to “Negative” class. The ground truth of the 20 generated videos was coded by a FACS coder. The human coder was told to classify each video into either “Positive” or “Negative”. As a result, 7 videos are “Positive” class and 13 videos are “Negative” class. Our FER algorithms are tested on these 20 generated videos. As the generated model is targeted in generating “Positive” and “Negative” classes, this study does not have “Neutral” as the ground truth in the test set. The accuracy of the FER methods are shown in Table 7.2. The results show that both methods perform well in detecting “Positive” class.

Qualitative Measurement

We further evaluate every emotion class and attempt to classify them into three simple

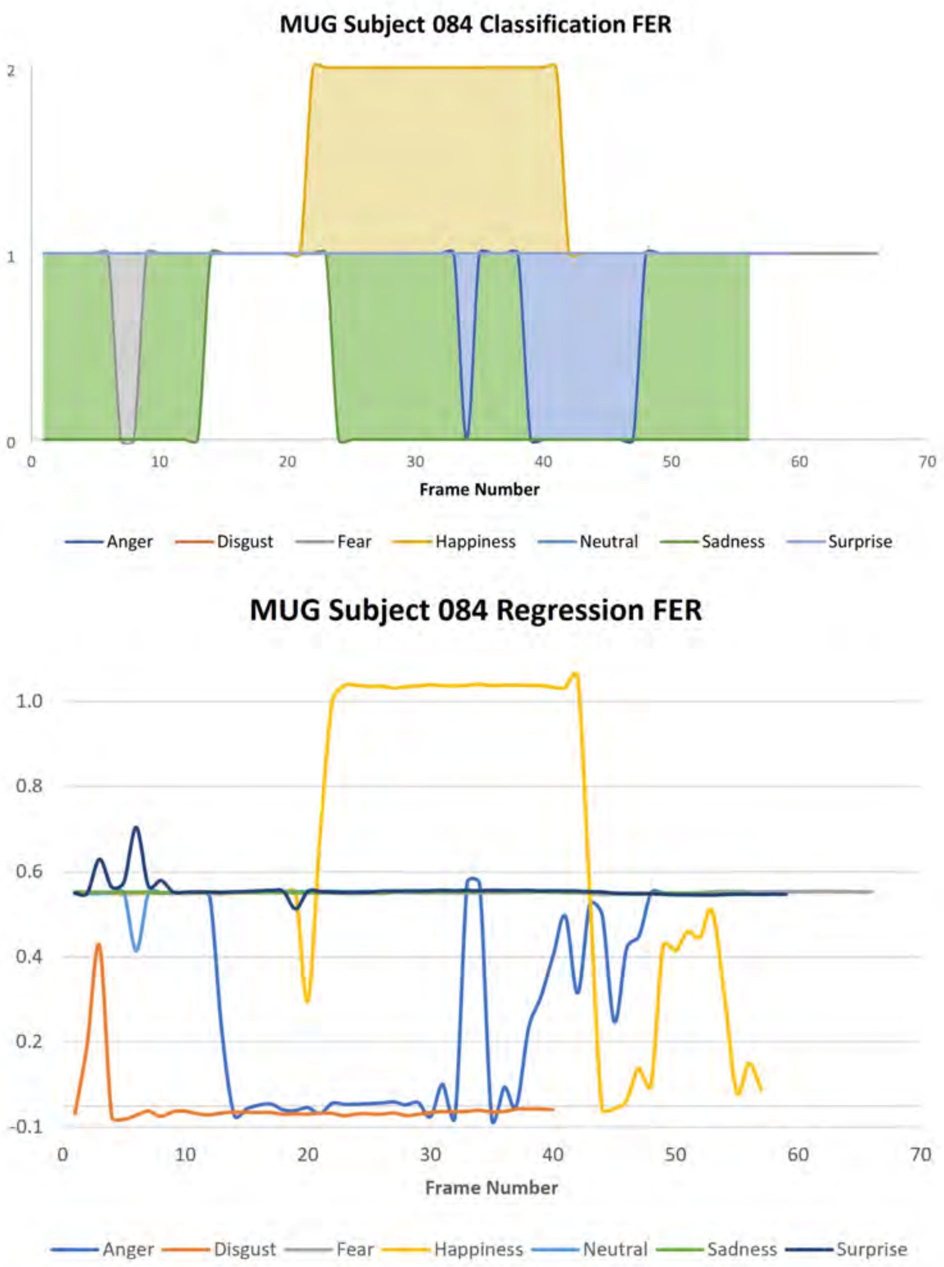


Figure 7.3: FER on both classification and regression tasks evaluated on subject 084 of MUG dataset. For classification task, "Negative" class is denoted as "0", "Neutral" class is denoted as "1" and "Positive" class is denoted as "2". For regression task, "Negative" class is denoted as "0", "Neutral" class is denoted as "0.5" and "Positive" class is denoted as "1.0". This task has continuous output values with intermediate values between each class. Both models show high agreement on "Happiness" where the "Positive" class are activated. Note: In classification, the coloured area under the curve is for visualisation only.

Table 7.2: Accuracy of FER Classification and Regression on generated data. The performance is evaluated compared to the emotion class annotated by human FACS coder. Both methods have identical performance and show better performance in "Positive" emotion class.

	Negative	Positive	Overall
FER Classification	0.46	0.57	0.50
FER Regression	0.46	0.71	0.55

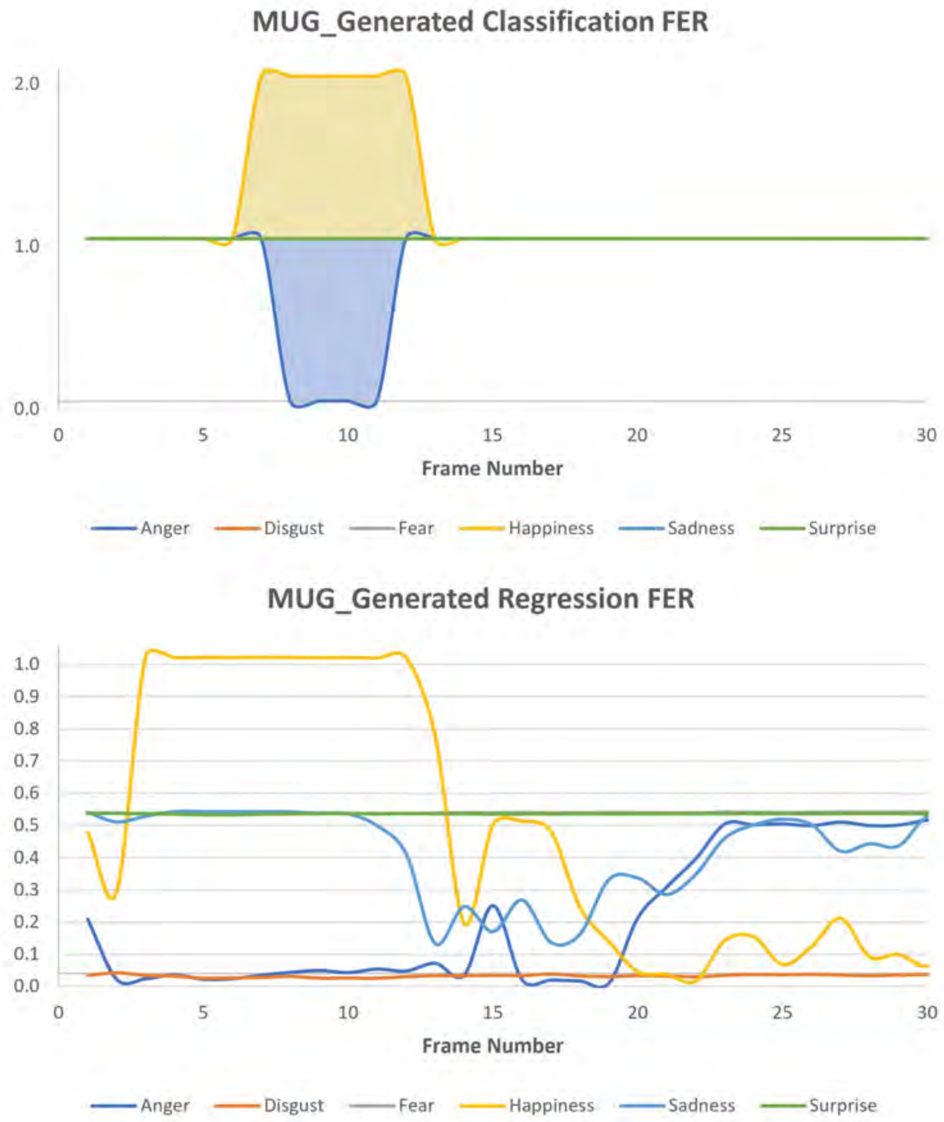
emotion classes. The temporal dynamic (onset and offset) is investigated as we are interested in knowing how well the algorithm deals with changing facial expression intensity.

For FER **classification** task, "happiness" and "anger" are identified. Other emotions are detected as "Neutral". As we can see, FER classification doesn't work well with sequences with temporal information. Our facial expression generation is fully trained using temporal information based on the current and one previous step, which may suggest both the classification model and sequence generation model uses different information for prediction and generation.

For FER **regression** task, "happiness" shows strong activation with clear onset and offset. "Anger" also shows partial agreement with FER classification. FER regression is capable of describing the temporal changes in facial expression more realistically as it has a continuous output. The recursive generation method requires temporal information in each step of the generation which works better when analysing with FER regression.

The recursive generation loop of our generation model exhibits Markov property which converges based on current and one previous step. For this to occur, the model must learn the temporal correlation between the frame sequences using the spatiotemporal information of the two frame pairs at each step of the recursive loop. Hence, temporal information is an important feature in this generation. FER classification has no intermediate values between each class. As the recursive generation progresses, temporal information is not involved in determining the activation of FER classification. This lack of temporal information makes the FER classification perform worse in evaluating our generated data. FER regression has higher accuracy and shows the onset and offset of facial expressions with fluctuations between the confidence of each expression. This is a more suitable method for facial expression sequence recognition as there are transitions between facial expressions.

Evaluation using Spontaneous Generated Facial Expression Sequences



MUG_Generated Classification FER

MUG_Generated Regression FER

Figure 7.4: FER on both classification and regression tasks evaluated on generated subjects. The recursive generation loop converges using Markov property. The temporal information is not captured by the classification model. While the regression model captured some of the transitions between the expression. Note: In classification, the coloured area under the curve is for visualisation only.

We also compare both FER tasks on expression sequence generated using Spontaneous Multi-Expression Sequence Generation of Figure 6.7. This video contains multiple changes in facial expressions. Similar to the previous evaluation, FER classification task does not capture the temporal information well while FER regression task captures some temporal transition between facial expressions. Nevertheless, both models also show high agreement on the "Positive" class generated at the end of the video.

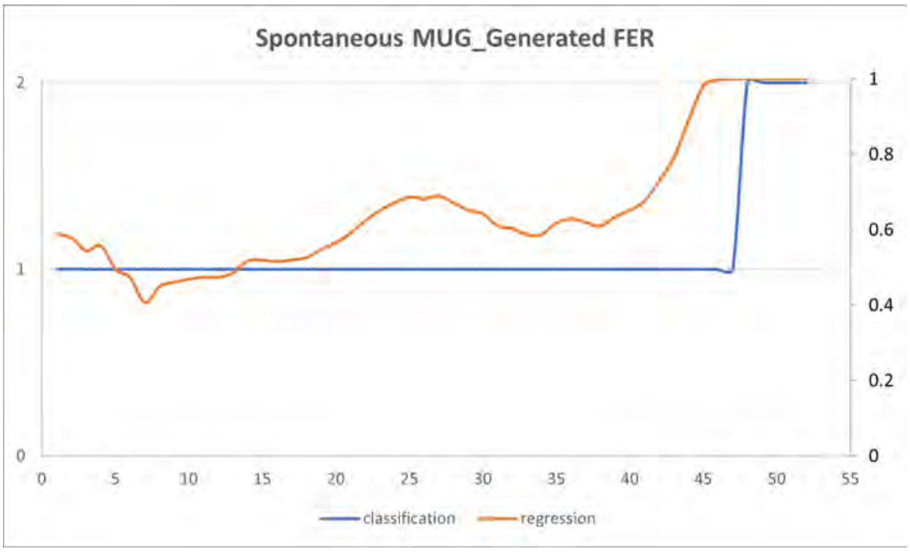


Figure 7.5: FER on both classification and regression tasks evaluated on spontaneous generated subjects (from Figure 6.7). Classification task does not take into account temporal information while regression does. Both models agree on the "Positive" class generated at the end of the video. Note: The y-axis on the left is for FER classification. The y-axis on the left is for FER regression.

Comparison of facial expression of different emotion classes cross-checked with FACS coder

As the onset and offset of MUG dataset was were annotated, human annotation (by a certified FACS coder) was used. The comparison are shown in Figure 7.6. We observed that both methods show agreement on "happiness" class. For "anger" class, FER regression outperforms FER classification where it shows higher agreement with human annotation in predicting the onset and offset of the expression.

Discussion

In both tasks, both models show high agreement on "happiness" where the "Positive" class are activated for every image sequence. A reason might be that about half of the training set is consisted of "happiness" images. "Happiness" involves large and distinct movements on AU6

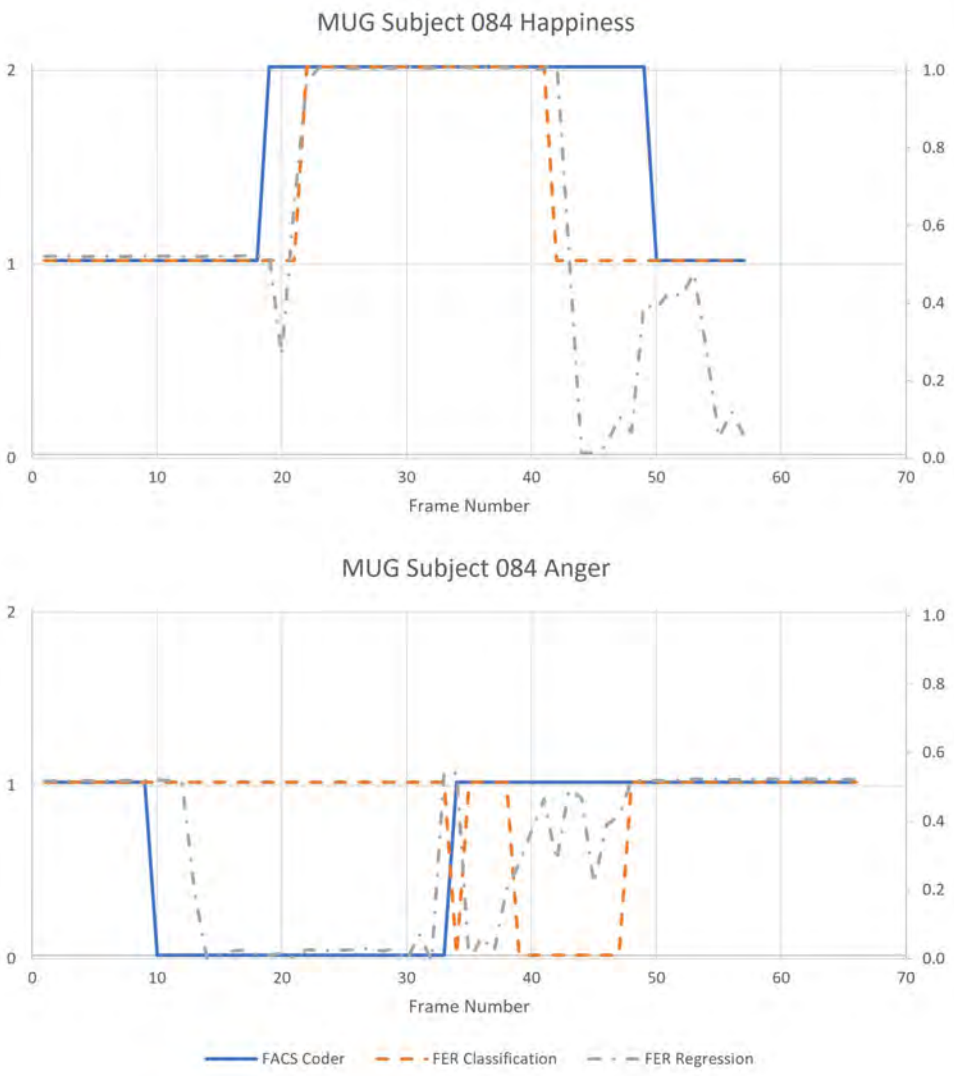


Figure 7.6: FER comparison between both methods and human annotation. “Positive” class is represented by “happiness” while “Negative” class is represented by “anger”. FER regression shows a more realistic transition of expression (where even human annotation is incapable of). FER regression shows higher agreement with human annotation relative to FER classification. Note: The y-axis on the left is for FACS Coder and FER classification. The y-axis on the left is for FER regression.

(cheek raiser) and AU12 (lip corner puller) that makes each phase (onset, apex, and offset) of the expression easily recognisable compared to other expressions.

Our model shows FER classification and regression accuracy of 0.85 and 0.80 when predicting real videos; 0.5 and 0.55 when predicting generated videos respectively. This shows a difference in the quality of our generated facial expression sequence compared to the real facial expression sequence. The higher accuracy in identifying “Positive” class may indicate that our generative model is better at generating facial expression sequences with “Positive” class.

Both FER regression and classification perform similarly in predicting “Negative” class; FER regression shows better performance in predicting “Positive” class. FER regression captures the temporal information of the change in facial expression which may be the reason for higher performance compared to FER classification.

In reality, facial expression is a sequence of facial movements. These sequence-based facial movements are known as dynamic expressions. Dynamic expressions contain temporal information of facial expressions which can improve the recognition performance of the model. With temporal information, it is also possible to anticipate facial expressions by looking into the first few sequences. We trained our model using static images of facial expressions and our regression approach shows our model has the ability to predict dynamic expressions with intermediate transitions between facial movements.

7.4 Conclusion

We demonstrate both classification and regression approaches on facial expression recognition by evaluating it on our facial expression sequence of real and generated data. FER classification and regression evaluated on real facial expression sequences show an accuracy of 0.80 which shows that our approach can be used in identifying facial expression sequences. However, when evaluated on generated facial expression sequences, the accuracy dropped to around 0.50. This is an indication that our generated sequence still has a quality difference compared to real facial expression. The “Positive” class predicted by FER regression in the generated data is 0.71 which shows that the generated “Positive” facial expression is more similar to a real facial expression compared to the “Negative” counterpart. Classification approach produces a discrete output and it is more suitable for application that requires exact or precise recognition of emotion. Regression approach creates a continuous output which can potentially be applied in classifying dynamic facial expression image sequences that changes over time. The onset, apex and offset phase of facial expression sequences can be predicted as it allows intermediate expression intensity between each prediction class. The transition between facial expression of spontaneous facial expression sequence can also be estimated. We evaluated both FER method on our original and generated data. We found that FER regression is more useful in analysing sequence data as the continuous output models the transitions between facial expressions better.

Chapter 8

Conclusion

8.1 Introduction

Facial expression can provide us with a glimpse into human emotion. Computer vision with deep learning approaches, such as convolutional neural networks, can potentially discover the patterns of facial expressions (both ME and MaE) by using a large number of input data. This can further analyse human emotions beyond conventional approaches.

For ME and MaE spotting, this thesis proposed a temporal oriented method that leverage the difference of duration ME and MaE by applying different frame skip based on each expression. Our method can spot ME and MaE simultaneously using end-to-end like deep learning. This model has state-of-the-art performance compared to the best candidate of MEGC2020.

For our generative method, we showed that style transfer can be used as an advanced image augmentation to generate more data. Next, more facial expression data can be produced by using latent representation as a computational efficient generation approach.

8.2 Research Findings

A summary of research objectives and each respective outcome is shown in Table 8.1.

The first objective is to identify research gap and determine research direction through literature review. We successfully identified a few key research gaps in this research which are a lack of dataset and method that can process ME and MaE simultaneously. These findings provide a general direction for the whole research project which is based on these two gaps.

The second objective is to address data deficiency by introducing a new ME long video dataset. We curated a new long video dataset, SAMM Long Videos, that contains ME and MaE. A baseline for the dataset on ME and MaE spotting was created using AU analysis of OpenFace.

The third objective is to implement an existing GAN model to generate new data and measure the quality of the generated data. This is done by performing reference-guided image synthesis by conducting style transfer on a pre-existing dataset, SAMM-LV. This method can be considered as an advanced image augmentation that changes the style of the subject without changing the facial movements. The facial movements transferred are compared with the original movements using correlation analysis on AUs and optical flow analysis.

The fourth objective is to propose a new ME detection algorithm based on deep learning method and validate it on ME long videos datasets. We noticed the main differences between ME and MaE and chose the duration difference of the expressions as our research interest. Based on the duration difference, we designed a two-stream convolutional neural network which is capable of spotting both expressions simultaneously.

The fifth objective is to design a new generative model that can generate spontaneous ME to overcome data deficiency and improve the diversity of datasets in ME research. A recursive generation model based on Restricted Boltzmann Machine was proposed and spontaneous generation of facial expressions can be conducted using this method. This method is a self-supervised method without any labels and is able to generate the entire facial expression sequence with only two input sequences (plus one constant input frame). This model can also be applied on ME generation.

Even though the majority of the objectives are met, there is still a lack of a fully end-to-end approach that can leverage deep learning to its full capability. The ME generation only involves small facial movements while not taking into account the full spontaneity of human movements (blinking or speaking).

Table 8.1: Objective and Outcome

No.	Objective	Outcome
1	To identify research gap and determine research direction through literature review.	Literature review on ME spotting and recognition especially on the state-of-the-art methods was thoroughly conducted.
2	To address data deficiency by introducing a new ME long video dataset.	SAMM Long Video dataset was curated based on the raw footage of SAMM. A baseline method using AU analysis of OpenFace was created.
3	To implement existing GAN models to generate new data and measure the quality of the generated data.	By using StarGANv2, we implement reference-guided image synthesis by conducting style transfer on a pre-existing dataset, SAMM-LV, as an advanced image augmentation to generate new data.
4	To propose a new ME detection algorithm based on deep learning method and validate it on ME long videos datasets.	A multi-stream network using 3D-CNN was trained and evaluated on our new ME dataset for ME and MaE spotting. This method can spots both expression simultaneously by leveraging the duration difference between both expression.
5	To design new generative model that can generate spontaneous ME to overcome data deficiency and improve the diversity of datasets in ME research.	We successfully generate spontaneous MaE using a sequence generation model. ME generation is also shown to be possible.

8.3 Future Works

The overall aim of this project is to improve the accuracy and generality of ME recognition and spotting in different experimental conditions. The training data will be further augmented so it represents real-world scenarios. The trained models need to be trained on a more diverse yet accurate dataset to further improve the results.

8.3.1 Fully end-to-end pipeline

To leverage the advantage of deep learning, fully end-to-end facial micro-expression should be considered as a foundation for this research. Currently, there is still a lack of dataset and end-to-end deep learning with a limited data pool has similar or slightly worse performance compared to methods that use traditional features as input.

8.3.2 Dataset number increment

The approaches in our thesis are only capable of generating realistic facial expressions in a fixed environment. The subtlety of ME generation is similar to real ME, however, other spontaneous facial movements are not generated (e.g. eye movements, blinks, mouth movements). To generate realistic long videos containing ME and MaE (similar to SAMM-LV), a combined pipeline for generating MaE and ME should be investigated.

8.3.3 Combination of all methods into one pipeline

Each separate part of this thesis can be a part of the full pipeline for micro-expression analysis. A possible full pipeline consisted of facial detection, ME spotting and proceed with ME recognition For application in the real world, the whole pipeline can function and complement each other to provide a better decision on the ME generated.

8.4 Concluding Remarks

This thesis presented a new approach to spotting ME and MaE simultaneously using the duration difference between these expressions. With the ability to spot both expressions simultaneously and show the state-of-the-art performance compared with the best candidate of MEGC2020, the contribution shows promising results and potential to be applied in real life. The facial expression sequence generation can be applied in ME generation which lays a foundation for future data generation methods.

References

[1] P. Ekman, *Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage.* Norton, 2001.

[2] M. Inc. “Face++ research toolkit.” (2013), [Online]. Available: `http://www.faceplusplus.com` (visited on 09/01/2021).

[3] T. Baltrušaitis, P. Robinson, and L.-P. Morency, “Openface: An open source facial behavior analysis toolkit,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2016, pp. 1–10.

[4] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, “Openface 2.0: Facial behavior analysis toolkit,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, IEEE, 2018, pp. 59–66.

[5] S.-T. Liong, Y. Gan, J. See, H.-Q. Khor, and Y.-C. Huang, “Shallow triple stream three-dimensional cnn (ststnet) for micro-expression recognition,” in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–5.

[6] P. Husák, J. Cech, and J. Matas, “Spotting facial micro-expressions ”in the wild”,” in *22nd Computer Vision Winter Workshop (Retz)*, 2017.

[7] D. Matsumoto and H. C. Hwang, “Microexpressions differentiate truths from lies about future malicious intent,” *Frontiers in psychology*, vol. 9, p. 2545, 2018.

[8] J. See, M. H. Yap, J. Li, X. Hong, and S.-J. Wang, “Megc 2019–the second facial micro-expressions grand challenge,” in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–5.

[9] S. Tomkins, *Affect imagery consciousness: Volume I: The positive affects.* Springer publishing company, 1962.

[10] P. Ekman, “Universals and cultural differences in facial expressions of emotion.,” in *Nebraska symposium on motivation*, University of Nebraska Press, 1971.

[11] A. Damasio and G. B. Carvalho, “The nature of feelings: Evolutionary and neurobiological origins,” *Nature reviews neuroscience*, vol. 14, no. 2, pp. 143–152, 2013.

[12] M. B. Arnold, “Emotion and personality.,” 1960.

[13] N. H. Frijda *et al.*, *The emotions.* Cambridge University Press, 1986.

[14] P. Ekman, W. V. Friesen, M. O’sullivan, A. Chan, I. Diacoyanni-Tarlatzis, K. Heider, R. Krause, W. A. LeCompte, T. Pitcairn, P. E. Ricci-Bitti, *et al.*, “Universals and cultural differences in the judgments of facial expressions of emotion.,” *Journal of personality and social psychology*, vol. 53, no. 4, p. 712, 1987.

[15] C. E. Izard and O. M. Haynes, “On the form and universality of the contempt expression: A challenge to ekman and friesen’s claim of discovery,” *Motivation and Emotion*, vol. 12, no. 1, pp. 1–16, 1988.

[16] C. Darwin and P. Prodger, *The expression of the emotions in man and animals.* Oxford University Press, USA, 1998.

[17] S. S. Tomkins, “Affect theory,” *Approaches to emotion*, vol. 163, no. 163-195, pp. 31–65, 1984.

[18] P. Ekman and W. V. Friesen, “Constants across cultures in the face and emotion.,” *Journal of personality and social psychology*, vol. 17, no. 2, p. 124, 1971.

[19] P. Ekman and D. Cordaro, “What is meant by calling emotions basic,” *Emotion review*, vol. 3, no. 4, pp. 364–370, 2011.

[20] W. E. Rinn, "The neuropsychology of facial expression: A review of the neurological and psychological mechanisms for producing facial expressions.," *Psychological bulletin*, vol. 95, no. 1, p. 52, 1984.

[21] W.-J. Yan, Q. Wu, J. Liang, Y.-H. Chen, and X. Fu, "How fast are the leaked facial expressions: The duration of micro-expressions," *Journal of Nonverbal Behavior*, vol. 37, no. 4, pp. 217–230, 2013.

[22] P. Ekman, "Darwin, deception, and facial expression," *Annals of the New York Academy of Sciences*, vol. 1000, no. 1, pp. 205–221, 2003.

[23] P. Ekman and G. Yamey, "Emotions revealed: Recognising facial expressions: In the first of two articles on how recognising faces and feelings can help you communicate, paul ekman discusses how recognising emotions can benefit you in your professional life," *Student BMJ*, vol. 12, pp. 140–142, 2004.

[24] P. Ekman, "Facial expression and emotion.," *American psychologist*, vol. 48, no. 4, p. 384, 1993.

[25] C. Pritsch, S. Telkemeyer, C. Mühlenbeck, and K. Liebal, "Perception of facial expressions reveals selective affect-biased attention in humans and orangutans," *Scientific reports*, vol. 7, no. 1, pp. 1–12, 2017.

[26] P. Ekman and W. V. Friesen, *Facial Action Coding System: Investigator's Guide.* Consulting Psychologists Press, 1978.

[27] X. Li, S. Cheng, Y. Li, M. Behzad, J. Shen, S. Zafeiriou, M. Pantic, and G. Zhao, "4dme: A spontaneous 4d micro-expression dataset with multimodalities," *IEEE Transactions on Affective Computing*, 2022.

[28] J. Sobotta, *Sobotta"s atlas and text-book of human anatomy*, 1909.

[29] X. Li, T. Pfister, X. Huang, G. Zhao, and M. Pietikainen, "A spontaneous micro-expression database: Inducement, collection and baseline," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, IEEE, 2013, pp. 1–6.

[30] F. Qu, S.-J. Wang, W.-J. Yan, H. Li, S. Wu, and X. Fu, “Cas (me)ˆ 2: A database for spontaneous macro-expression and micro-expression spotting and recognition,” *IEEE Transactions on Affective Computing*, 2017.

[31] C. H. Yap, C. Kendrick, and M. H. Yap, “Samm long videos: A spontaneous facial micro-and macro-expressions dataset,” in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, IEEE, 2020, pp. 771–776.

[32] S. Polikovsky, Y. Kameda, and Y. Ohta, “Facial micro-expressions recognition using high speed camera and 3d-gradient descriptor,” in *Crime Detection and Prevention (ICDP 2009), 3rd International Conference on*, IET, 2009, pp. 1–6.

[33] G. Warren, E. Schertler, and P. Bull, “Detecting deception from emotional and unemotional cues,” *Journal of Nonverbal Behavior*, vol. 33, no. 1, pp. 59–69, 2009.

[34] M. Shreve, S. Godavarthy, D. Goldgof, and S. Sarkar, “Macro- and micro-expression spotting in long videos using spatio-temporal strain,” in *2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011)*, 2011, pp. 51–56. DOI: 10.1109/FG.2011.5771451.

[35] W.-J. Yan, Q. Wu, Y.-J. Liu, S.-J. Wang, and X. Fu, “Casme database: A dataset of spontaneous micro-expressions collected from neutralized faces,” in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, IEEE, 2013, pp. 1–7.

[36] W.-J. Yan, X. Li, S.-J. Wang, G. Zhao, Y.-J. Liu, Y.-H. Chen, and X. Fu, “Casme ii: An improved spontaneous micro-expression database and the baseline evaluation,” *PloS one*, vol. 9, no. 1, 2014.

[37] A. K. Davison, C. Lansley, N. Costen, K. Tan, and M. H. Yap, “Samm: A spontaneous micro-facial movement dataset,” *IEEE Transactions on Affective Computing*, vol. 9, no. 1, pp. 116–129, 2016.

[38] T.-K. Tran, Q.-N. Vo, X. Hong, X. Li, and G. Zhao, “Micro-expression spotting: A new benchmark,” *Neurocomputing*, vol. 443, pp. 356–368, 2021.

[39] X. Ben, Y. Ren, J. Zhang, S.-J. Wang, K. Kpalma, W. Meng, and Y.-J. Liu, "Video-based facial micro-expression analysis: A survey of datasets, features and algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[40] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[41] N. Aifanti, C. Papachristou, and A. Delopoulos, "The mug facial expression database," in *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, IEEE, 2010, pp. 1–4.

[42] M. Valstar and M. Pantic, "Induced disgust, happiness and surprise: An addition to the mmi facial expression database," in *Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, Paris, France, 2010, p. 65.

[43] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[44] S. Polikovsky and Y. Kameda, "Facial micro-expression detection in hi-speed video based on facial action coding system (facs)," *IEICE transactions on information and systems*, vol. 96, no. 1, pp. 81–92, 2013.

[45] M. Chen, H. T. Ma, J. Li, and H. Wang, "Emotion recognition using fixed length micro-expressions sequence and weighting method," in *Real-time Computing and Robotics (RCAR), IEEE International Conference on*, IEEE, 2016, pp. 427–430.

[46] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[47] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[48] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *Pattern Analysis and Machine Intelligence,*

IEEE Transactions on, vol. 29, no. 6, pp. 915–928, 2007, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1110.

[49] Y. Guo, Y. Tian, X. Gao, and X. Zhang, “Micro-expression recognition based on local binary patterns from three orthogonal planes and nearest neighbor method,” in *Neural Networks (IJCNN), 2014 International Joint Conference on*, IEEE, 2014, pp. 3473–3479.

[50] B. B. Talukder, B. Chowdhury, T. Howlader, and S. M. Rahman, “Intelligent recognition of spontaneous expression using motion magnification of spatio-temporal data,” in *Pacific-Asia Workshop on Intelligence and Security Informatics*, Springer, 2016, pp. 114–128.

[51] X. Zhu, X. Ben, S. Liu, R. Yan, and W. Meng, “Coupled source domain targetized with updating tag vectors for micro-expression recognition,” *Multimedia Tools and Applications*, vol. 77, no. 3, pp. 3105–3124, 2018.

[52] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[53] B. D. Lucas, T. Kanade, *et al.*, “An iterative image registration technique with an application to stereo vision,” 1981.

[54] S.-T. Liong, J. See, K. Wong, and R. C.-W. Phan, “Less is more: Micro-expression recognition from video using apex frame,” *Signal Processing: Image Communication*, vol. 62, pp. 82–92, 2018.

[55] S.-T. Liong, J. See, R. C.-W. Phan, Y.-H. Oh, A. C. Le Ngo, K. Wong, and S.-W. Tan, “Spontaneous subtle expression detection and recognition based on facial strain,” *Signal Processing: Image Communication*, vol. 47, pp. 170–182, 2016.

[56] H. Zheng, X. Geng, and Z. Yang, “A relaxed k-svd algorithm for spontaneous micro-expression recognition,” in *Pacific Rim International Conference on Artificial Intelligence*, Springer, 2016, pp. 692–699.

[57] S. Zhang, B. Feng, Z. Chen, and X. Huang, “Micro-expression recognition by aggregating local spatio-temporal patterns,” in *International Conference on Multimedia Modeling*, Springer, 2017, pp. 638–648.

[58] Q. Li, J. Yu, T. Kurihara, and S. Zhan, “Micro-expression analysis by fusing deep convolutional neural network and optical flow,” in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, IEEE, 2018, pp. 265–270.

[59] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[60] M. Buhari, C. Ooi, V. Baskaran, and W. Tan, “Motion and geometric feature analysis for real-time automatic micro-expression recognition systems,” 2021.

[61] V. Mayya, R. M. Pai, and M. M. Pai, “Combining temporal interpolation and dcnn for faster recognition of micro-expressions in video sequences,” in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2016, pp. 699–703.

[62] S.-J. Wang, B.-J. Li, Y.-J. Liu, W.-J. Yan, X. Ou, X. Huang, F. Xu, and X. Fu, “Micro-expression recognition with small sample size by transferring long-term convolutional neural network,” *Neurocomputing*, vol. 312, pp. 251–262, 2018.

[63] J. Li, Y. Wang, J. See, and W. Liu, “Micro-expression recognition based on 3d flow convolutional neural network,” *Pattern Analysis and Applications*, vol. 22, no. 4, pp. 1331–1339, 2019.

[64] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman, “Eulerian video magnification for revealing subtle changes in the world,” 2012.

[65] Y. Wang, J. See, Y.-H. Oh, R. C.-W. Phan, Y. Rahulamathavan, H.-C. Ling, S.-W. Tan, and X. Li, “Effective recognition of facial micro-expressions with video motion magnification,” *Multimedia Tools and Applications*, pp. 1–26, 2016.

[66] Y. Liu, H. Du, L. Zheng, and T. Gedeon, "A neural micro-expression recognizer," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–4.

[67] Z. Xia, H. Liang, X. Hong, and X. Feng, "Cross-database micro-expression recognition with deep convolutional networks," in *Proceedings of the 2019 3rd International Conference on Biometric Engineering and Applications*, ACM, 2019, pp. 56–60.

[68] Z. Xia, X. Feng, X. Hong, and G. Zhao, "Spontaneous facial micro-expression recognition via deep convolutional network," in *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, IEEE, 2018, pp. 1–6.

[69] Z. Xia, X. Hong, X. Gao, X. Feng, and G. Zhao, "Spatiotemporal recurrent convolutional networks for recognizing spontaneous micro-expressions," *IEEE Transactions on Multimedia*, 2019.

[70] Y. Zong, X. Huang, W. Zheng, Z. Cui, and G. Zhao, "Learning from hierarchical spatiotemporal descriptors for micro-expression recognition," *IEEE Transactions on Multimedia*, 2018.

[71] C. Hu, D. Jiang, H. Zou, X. Zuo, and Y. Shu, "Multi-task micro-expression recognition combining deep and handcrafted features," in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 946–951.

[72] M. Peng, C. Wang, T. Chen, G. Liu, and X. Fu, "Dual temporal scale convolutional neural network for micro-expression recognition," *Frontiers in psychology*, vol. 8, p. 1745, 2017.

[73] H.-Q. Khor, J. See, R. C. W. Phan, and W. Lin, "Enriched long-term recurrent convolutional network for facial micro-expression recognition," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, IEEE, 2018, pp. 667–674.

[74] H.-Q. Khor, J. See, S.-T. Liong, R. C. Phan, and W. Lin, "Dual-stream shallow networks for facial micro-expression recognition," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 36–40.

[75] L. Zhou, Q. Mao, and L. Xue, “Dual-inception network for cross-database micro-expression recognition,” in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–5.

[76] K. Li, Y. Zong, B. Song, J. Zhu, J. Shi, W. Zheng, and L. Zhao, “Three-stream convolutional neural network for micro-expression recognition,” in *Proc. 26th Int. Conf. Neural Inf. Process.(ICONIP)*, 2019.

[77] Y. Gan and S.-T. Liong, “Bi-directional vectors from apex in cnn for micro-expression recognition,” in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, IEEE, 2018, pp. 168–172.

[78] Y. Gan, S.-T. Liong, W.-C. Yau, Y.-C. Huang, and L.-K. Tan, “Off-apexnet on micro-expression recognition system,” *Signal Processing: Image Communication*, vol. 74, pp. 129–139, 2019.

[79] R. Belaiche, Y. Liu, C. Migniot, D. Ginhac, and F. Yang, “Cost-effective cnns for real-time micro-expression recognition,” *Applied Sciences*, vol. 10, no. 14, p. 4959, 2020.

[80] M. Verma, S. K. Vipparthi, G. Singh, and S. Murala, “Learnet: Dynamic imaging network for micro expression recognition,” *IEEE Transactions on Image Processing*, vol. 29, pp. 1618–1627, 2019.

[81] M. Aouayeb, W. Hamidouche, K. Kpalma, and A. Benazza-Benyahia, “A spatiotemporal deep learning solution for automatic micro-expressions recognition from local facial regions,” in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2019, pp. 1–6.

[82] S. P. T. Reddy, S. T. Karri, S. R. Dubey, and S. Mukherjee, “Spontaneous facial micro-expression recognition using 3d spatiotemporal convolutional neural networks,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–8.

[83] T. Huang, L. Chen, Y. Feng, X. Ben, R. Xiao, and T. Xue, “A multiview representation framework for micro-expression recognition,” *IEEE Access*, vol. 7, pp. 120 670–120 680, 2019.

[84] R. Zhi and M. Wan, “Dynamic facial expression feature learning based on sparse rnn,” in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, IEEE, 2019, pp. 1373–1377.

[85] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models-their training and application,” *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[86] M. Peng, Z. Wu, Z. Zhang, and T. Chen, “From macro to micro expression recognition: Deep learning on small datasets using transfer learning,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, IEEE, 2018, pp. 657–661.

[87] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[88] N. Van Quang, J. Chun, and T. Tokuyama, “Capsulenet for micro-expression recognition,” in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–7.

[89] S.-T. Liong, J. See, K. Wong, and R. C.-W. Phan, “Automatic micro-expression recognition from long video using a single spotted apex,” in *Asian conference on computer vision*, Springer, 2016, pp. 345–360.

[90] Y. Li, X. Huang, and G. Zhao, “Can micro-expression be recognized based on single apex frame?” In *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 3094–3098.

[91] Z. Zhang, T. Chen, H. Meng, G. Liu, and X. Fu, “Smeconvnet: A convolutional neural network for spotting spontaneous facial micro-expression from long videos,” *IEEE Access*, vol. 6, pp. 71 143–71 151, 2018.

[92] A. K. Davison, M. H. Yap, and C. Lansley, “Micro-facial movement detection using individualised baselines and histogram-based descriptors,” in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, IEEE, 2015, pp. 1864–1869.

[93] A. K. Davison, “Micro-facial movement detection using spatio-temporal features,” Ph.D. dissertation, Manchester Metropolitan University, 2016.

[94] A. Moilanen, G. Zhao, and M. Pietikainen, “Spotting rapid facial movements from videos using appearance-based feature difference analysis,” in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug. 2014, pp. 1722–1727. DOI: 10.1109/ICPR.2014.303.

[95] S.-T. Liong, J. See, K. Wong, A. C. Le Ngo, Y.-H. Oh, and R. Phan, “Automatic apex frame spotting in micro-expression database,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, IEEE, 2015, pp. 665–669.

[96] D. Patel, G. Zhao, and M. Pietikäinen, “Spatiotemporal integration of optical flow vectors for micro-expression detection,” in *Advanced Concepts for Intelligent Vision Systems*, Springer, 2015, pp. 369–380.

[97] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, “Learning a no-reference quality metric for single-image super-resolution,” *Computer Vision and Image Understanding*, vol. 158, pp. 1–16, 2017.

[98] V. Esmaeili and S. O. Shahdi, “Automatic micro-expression apex spotting using cubic-lbp,” *Multimedia Tools and Applications*, pp. 1–19, 2020.

[99] T.-K. Tran, Q.-N. Vo, X. Hong, and G. Zhao, “Dense prediction for micro-expression spotting based on deep sequence model,” *Electronic Imaging*, vol. 2019, no. 8, pp. 401–1, 2019.

[100] L.-w. Zhang, J. Li, S. Wang, X. Duan, W. Yan, H. Xie, and S. Huang, “Spatio-temporal fusion for macro-and micro-expression spotting in long video sequences,” in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, 2020, pp. 245–252.

[101] W.-W. Yu, J. Jiang, and Y.-J. Li, “Lssnet: A two-stream convolutional neural network for spotting macro-and micro-expression in long videos,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4745–4749.

[102] M. Verburg and V. Menkovski, "Micro-expression detection in long videos using optical flow and recurrent neural networks," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–6.

[103] B. Sun, S. Cao, J. He, and L. Yu, "Two-stream attention-aware network for spontaneous micro-expression movement spotting," in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, 2019, pp. 702–705.

[104] D. Borza, R. Itu, and R. Danescu, "Real-time micro-expression detection from high speed cameras," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, IEEE, 2017, pp. 357–361.

[105] B. Yang, J. Wu, Z. Zhou, M. Komiya, K. Kishimoto, J. Xu, K. Nonaka, T. Horiuchi, S. Komorita, G. Hattori, *et al.*, "Facial action unit-based deep learning framework for spotting macro-and micro-expressions in long video sequences," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4794–4798.

[106] S.-J. Wang, Y. He, J. Li, and X. Fu, "Mesnet: A convolutional neural network for spotting multi-scale micro-expression intervals in long videos," *IEEE Transactions on Image Processing*, vol. 30, pp. 3956–3969, 2021.

[107] H. Pan, L. Xie, and Z. Wang, "Local bilinear convolutional neural network for spotting macro- and micro-expression intervals in long video sequences," in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, 2020, pp. 343–347.

[108] D. A. Chanti and A. Caplier, "Ads-me: Anomaly detection system for micro-expression spotting," *arXiv preprint arXiv:1903.04354*, 2019.

[109] C. H. Yap, M. H. Yap, A. Davison, C. Kendrick, J. Li, S.-J. Wang, and R. Cunningham, "3d-cnn for facial micro-and macro-expression spotting on long video sequences using temporal oriented reference frame," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 7016–7020.

[110] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 818–833.

[111] Y. Liu, X. Hou, J. Chen, C. Yang, G. Su, and W. Dou, "Facial expression recognition and generation using sparse autoencoder," in *2014 International Conference on Smart Computing*, IEEE, 2014, pp. 125–130.

[112] L. Fan, W. Huang, C. Gan, J. Huang, and B. Gong, "Controllable image-to-video translation: A case study on facial expression generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3510–3517.

[113] Y.-C. Chen, X. Xu, Z. Tian, and J. Jia, "Homomorphic latent space interpolation for unpaired image-to-image translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2408–2416.

[114] Y. Deng, J. Yang, D. Chen, F. Wen, and X. Tong, "Disentangled and controllable face image generation via 3d imitative-contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5154–5163.

[115] H. Ding, K. Sricharan, and R. Chellappa, "Exprgan: Facial expression editing with controllable expression intensity," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[116] G. Shen, W. Huang, C. Gan, M. Tan, J. Huang, W. Zhu, and B. Gong, "Facial image-to-video translation by a hidden affine transformation," in *Proceedings of the 27th ACM international conference on Multimedia*, 2019, pp. 2505–2513.

[117] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," in *Advances in Neural Information Processing Systems*, 2019, pp. 7135–7145.

[118] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 613–621.

[119] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2830–2839.

[120] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535.

[121] J. Li, M. H. Yap, W.-H. Cheng, J. See, X. Hong, X. Li, and S.-J. Wang, "Fme'21: 1st workshop on facial micro-expression: Advanced techniques for facial expressions generation and spotting," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 5700–5701.

[122] Y. Zhang, Y. Zhao, Y. Wen, Z. Tang, X. Xu, and M. Liu, "Facial prior based first order motion model for micro-expression generation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4755–4759.

[123] Y. Xu, S. Zhao, H. Tang, X. Mao, T. Xu, and E. Chen, "Famgan: Fine-grained aus modulation based generative adversarial network for micro-expression generation," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4813–4817.

[124] X. Fan, A. R. Shahid, and H. Yan, "Facial micro-expression generation based on deep motion retargeting and transfer learning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4735–4739.

[125] H.-X. Xie, L. Lo, H.-H. Shuai, and W.-H. Cheng, "Au-assisted graph attention convolutional network for micro-expression recognition," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2871–2880.

[126] J. Yu, C. Zhang, Y. Song, and W. Cai, "Ice-gan: Identity-aware and capsule-enhanced gan for micro-expression recognition and synthesis," *arXiv preprint arXiv:2005.04370*, 2020.

[127] C. H. Yap, R. Cunningham, A. K. Davison, and M. H. Yap, “Synthesising facial macro- and micro-expressions using reference guided style transfer,” *Journal of Imaging*, vol. 7, no. 8, p. 142, 2021.

[128] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8188–8197.

[129] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Training models of shape from sets of examples,” in *BMVC92*, Springer, 1992, pp. 9–18.

[130] M. Pilch, Y. Wenner, E. Strohmayr, M. Preising, C. Friedburg, E. M. Zu Bexten, B. Lorenz, and K. Stieger, “Automated segmentation of retinal blood vessels in spectral domain optical coherence tomography scans,” *Biomedical optics express*, vol. 3, no. 7, pp. 1478–1491, 2012.

[131] T. Baltrusaitis, P. Robinson, and L.-P. Morency, “Constrained local neural fields for robust facial landmark detection in the wild,” in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 354–361.

[132] A. Zadeh, Y. Chong Lim, T. Baltrusaitis, and L.-P. Morency, “Convolutional experts constrained local model for 3d facial landmark detection,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2519–2528.

[133] D. Cristinacce, T. F. Cootes, *et al.*, “Feature detection and tracking with constrained local models.,” in *Bmvc*, Citeseer, vol. 1, 2006, p. 3.

[134] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, “Style aggregated network for facial landmark detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 379–388.

[135] B. Amos, B. Ludwiczuk, M. Satyanarayanan, *et al.*, “Openface: A general-purpose face recognition library with mobile applications,” *CMU School of Computer Science*, vol. 6, p. 2, 2016.

[136] T. Baltrušaitis, M. Mahmoud, and P. Robinson, “Cross-dataset learning and person-specific normalisation for automatic action unit detection,” in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, IEEE, vol. 6, 2015, pp. 1–6.

[137] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” In *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, pp. 2146–2153.

[138] D. Sage and M. Unser, “Teaching image-processing programming in java,” *IEEE Signal Processing Magazine*, vol. 20, no. 6, pp. 43–52, 2003.

[139] T. Chen, “Adaptive temporal interpolation using bidirectional motion estimation and compensation,” in *Proceedings. International Conference on Image Processing*, IEEE, vol. 2, 2002, pp. II–II.

[140] T. Pfister, X. Li, G. Zhao, and M. Pietikäinen, “Recognising spontaneous facial micro-expressions,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 1449–1456.

[141] M. Elgharib, M. Hefeeda, F. Durand, and W. T. Freeman, “Video magnification in presence of large motions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4119–4127.

[142] N. Wadhwa, M. Rubinstein, F. Durand, and W. T. Freeman, “Phase-based video motion processing,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–10, 2013.

[143] ——, “Riesz pyramids for fast phase-based video magnification,” in *2014 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2014, pp. 1–10.

[144] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Scandinavian conference on Image analysis*, Springer, 2003, pp. 363–370.

[145] J.-Y. Bouguet *et al.*, “Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm,” *Intel corporation*, vol. 5, no. 1-10, p. 4, 2001.

[146] C. Zach, T. Pock, and H. Bischof, “A duality based approach for realtime tv-l 1 optical flow,” in *Joint pattern recognition symposium*, Springer, 2007, pp. 214–223.

[147] J. S. Pérez, E. Meinhardt-Llopis, and G. Facciolo, “Tv-l1 optical flow estimation,” *Image Processing On Line*, vol. 2013, pp. 137–150, 2013.

[148] A. Romero, J. León, and P. Arbeláez, “Multi-view dynamic facial action unit detection,” *Image and Vision Computing*, 2018.

[149] S.-J. Wang, W.-J. Yan, G. Zhao, X. Fu, and C.-G. Zhou, “Micro-expression recognition using robust principal component analysis and local spatiotemporal directional features,” in *European Conference on computer vision*, Springer, 2014, pp. 325–338.

[150] P. Smolensky, “Information processing in dynamical systems: Foundations of harmony theory,” Colorado Univ at Boulder Dept of Computer Science, Tech. Rep., 1986.

[151] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[152] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[153] G. E. Hinton, “Boltzmann machine,” *Scholarpedia*, vol. 2, no. 5, p. 1668, 2007.

[154] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, “Generating 3d faces using convolutional mesh autoencoders,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 704–720.

[155] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville, “Adversarially learned inference,” in *International Conference on Learning Representations*, 2017. [Online]. Available: `https://openreview.net/forum?id=B1ElR4cgg`.

[156] A. Van Den Oord, O. Vinyals, *et al.*, “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, 2017.

[157] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra, “Deep autoregressive networks,” in *International Conference on Machine Learning*, PMLR, 2014, pp. 1242–1250.

[158] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[159] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[160] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, “End-to-end, single-stream temporal action detection in untrimmed videos,” in *Procedings of the British Machine Vision Conference 2017*, British Machine Vision Association, 2019.

[161] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, “End-to-end learning of action detection from frame glimpses in videos,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2678–2687.

[162] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” *Advances in neural information processing systems*, vol. 27, 2014.

[163] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, “Temporal segment networks: Towards good practices for deep action recognition,” in *European conference on computer vision*, Springer, 2016, pp. 20–36.

[164] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “Slowfast networks for video recognition,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[165] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, “Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition,” *Signal Processing: Image Communication*, vol. 71, pp. 76–87, 2019.

[166] L. Wang, L. Ge, R. Li, and Y. Fang, “Three-stream cnns for action recognition,” *Pattern Recognition Letters*, vol. 92, pp. 33–40, 2017.

[167] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *arXiv preprint arXiv:1508.06576*, 2015.

[168] C. Li and M. Wand, “Combining markov random fields and convolutional neural networks for image synthesis,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2479–2486.

[169] A. J. Champandard, “Semantic style transfer and turning two-bit doodles into fine artworks,” *arXiv preprint arXiv:1603.01768*, 2016.

[170] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand, “Style transfer for headshot portraits,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–14, 2014.

[171] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[172] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[173] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[174] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[175] A. Jaiswal, W. AbdAlmageed, Y. Wu, and P. Natarajan, “Capsulegan: Generative adversarial capsule network,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[176] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” in *Advances in neural information processing systems*, 2004, pp. 321–328.

[177] N. Otberdout, M. Daoudi, A. Kacem, L. Ballihi, and S. Berretti, “Dynamic facial expression generation on hilbert hypersphere with conditional wasserstein generative adversarial nets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[178] X. Ning, S. Xu, Y. Zong, W. Tian, L. Sun, and X. Dong, “Emotiongan: Facial expression synthesis based on pre-trained generator,” in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1518, 2020, p. 012 031.

[179] J. Ling, H. Xue, L. Song, S. Yang, R. Xie, and X. Gu, “Toward fine-grained facial expression manipulation,” in *European Conference on Computer Vision*, Springer, 2020, pp. 37–53.

[180] F. Wang, S. Xiang, T. Liu, and Y. Fu, “Attention based facial expression manipulation,” in *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, 2021, pp. 1–6.

[181] Y. Zhang, R. Liu, Y. Pan, D. Wu, Y. Zhu, and Z. Bai, “Gi-aee: Gan inversion based attentive expression embedding network for facial expression editing,” in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2021, pp. 2453–2457.

[182] J. Li, S. Wang, M. H. Yap, J. See, X. Hong, and X. Li, “Megc2020-the third facial micro-expression grand challenge,” in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)(FG)*, pp. 234–237.

[183] D. Freedman, R. Pisani, and R. Purves, “Statistics (international student edition),” *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.

[184] R. A. Fisher, “On the’probable error’of a coefficient of correlation deduced from a small sample,” *Metron*, vol. 1, pp. 1–32, 1921.

[185] R. W. Schafer, “What is a savitzky-golay filter?[lecture notes],” *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[186] A. Savitzky and M. J. Golay, “Smoothing and differentiation of data by simplified least squares procedures.,” *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[187] H. Matsuura, *What is power spectral density function?* 2017. [Online]. Available: `https://www.cygres.com/OcnPageE/Glosry/SpecE.html`.

[188] S. Butterworth *et al.*, “On the theory of filter amplifiers,” *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.

[189] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[190] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[191] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[192] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[193] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.

[194] B. M. Hewitt, M. H. Yap, E. F. Hodson-Tole, A. J. Kennerley, P. S. Sharp, and R. A. Grant, “A novel automated rodent tracker (art), demonstrated in a mouse model of amyotrophic lateral sclerosis,” *Journal of neuroscience methods*, vol. 300, pp. 147–156, 2018.

[195] I. Daubechies, “The wavelet transform, time-frequency localization and signal analysis,” *IEEE transactions on information theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[196] J. Li, C. Soladie, R. Seguier, S.-J. Wang, and M. H. Yap, “Spotting micro-expressions on long videos sequences,” in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, IEEE, 2019, pp. 1–5.

[197] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn, “Disfa: A spontaneous facial action intensity database,” *IEEE Transactions on Affective Computing*, vol. 4, no. 2, pp. 151–160, 2013.

[198] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder, “The semaine database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent,” *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 5–17, 2012.

[199] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard, “Bp4d-spontaneous: A high-resolution spontaneous 3d dynamic facial expression database,” *Image and Vision Computing*, vol. 32, no. 10, pp. 692–706, 2014.

[200] Y. He, S.-J. Wang, J. Li, and M. H. Yap, “Spotting macro-and micro-expression intervals in long video sequences,” *arXiv preprint arXiv:1912.11985*, 2019.

[201] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[202] Y.-H. Oh, J. See, A. C. Le Ngo, R. C.-W. Phan, and V. M. Baskaran, “A survey of automatic facial micro-expression analysis: Databases, methods, and challenges,” *Frontiers in psychology*, vol. 9, p. 1128, 2018.

[203] B. F. Klare, M. J. Burge, J. C. Klontz, R. W. V. Bruegge, and A. K. Jain, “Face recognition performance: Role of demographic information,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1789–1801, 2012.

[204] J. Buolamwini and T. Gebru, “Gender shades: Intersectional accuracy disparities in commercial gender classification,” in *Conference on fairness, accountability and transparency*, PMLR, 2018, pp. 77–91.

[205] R. V. Garcia, L. Wandzik, L. Grabner, and J. Krueger, “The harms of demographic bias in deep face recognition research,” in *2019 International Conference on Biometrics (ICB)*, IEEE, 2019, pp. 1–6.

[206] M. Bertero, T. A. Poggio, and V. Torre, “Ill-posed problems in early vision,” *Proceedings of the IEEE*, vol. 76, no. 8, pp. 869–889, 1988.

[207] P. Turaga, R. Chellappa, and A. Veeraraghavan, “Advances in video-based human activity analysis: Challenges and approaches,” in *Advances in Computers*, vol. 80, Elsevier, 2010, pp. 237–290.

[208] S. Lyu and E. P. Simoncelli, “Nonlinear image representation using divisive normalization,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[209] A. Halevy, P. Norvig, and F. Pereira, “The unreasonable effectiveness of data,” *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.

[210] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, “Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning,” *Nature neuroscience*, vol. 21, no. 9, pp. 1281–1289, 2018.

[211] J. Li, Z. Dong, S. Lu, S.-J. Wang, W.-J. Yan, Y. Ma, Y. Liu, C. Huang, and X. Fu, “Cas (me) 3: A third generation facial spontaneous micro-expression database with depth information and high ecological validity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[212] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[213] Q. Memon, M. Ahmed, S. Ali, A. R. Memon, and W. Shah, “Self-driving and driver relaxing vehicle,” in *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, IEEE, 2016, pp. 170–174.

[214] P. A. Hancock, I. Nourbakhsh, and J. Stewart, "On the future of transportation in an era of automated and autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7684–7691, 2019.

[215] X. Wang, X. Wang, and Y. Ni, "Unsupervised domain adaptation for facial expression recognition using generative adversarial networks," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[216] J. Kossaifi, L. Tran, Y. Panagakis, and M. Pantic, "Gagan: Geometry-aware generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 878–887.

[217] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, "Latent video transformer," in *VISIGRAPP 2021-Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2021, pp. 101–112.

[218] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021.

[219] G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1025–1032.

[220] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[221] O. Woodford, "Notes on contrastive divergence," *Department of Engineering Science, University of Oxford, Tech. Rep*, 2006.

[222] J. Kim and S. Lee, "Deep learning of human visual sensitivity in image quality assessment framework," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1676–1684.

[223] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, “Vivit: A video vision transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6836–6846.

[224] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, “Animating arbitrary objects via deep motion transfer,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2377–2386.

[225] N. Srivastava, E. Mansimov, and R. Salakhudinov, “Unsupervised learning of video representations using lstms,” in *International conference on machine learning*, PMLR, 2015, pp. 843–852.

[226] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, IEEE, 2010, pp. 94–101.

[227] D. Lundqvist, A. Flykt, and A. hman, “The karolinska directed emotional faces,” 1998.

[228] A. Mollahosseini, B. Hasani, and M. H. Mahoor, “Affectnet: A database for facial expression, valence, and arousal computing in the wild,” *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18–31, 2017.

[229] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[230] W. Deng, Y. Fang, Z. Xu, and J. Hu, “Facial landmark localization by enhanced convolutional neural network,” *Neurocomputing*, vol. 273, pp. 222–229, 2018.

[231] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.

[232] P. Bian, Z. Xie, and Y. Jin, “Multi-task feature learning-based improved supervised descent method for facial landmark detection,” *Signal, Image and Video Processing*, vol. 12, no. 1, pp. 17–24, 2018.

[233] E. B. Fowlkes and C. L. Mallows, “A method for comparing two hierarchical clusterings,” *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[234] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, “Supervised machine learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[235] Y. Freund and R. E. Schapire, “Large margin classification using the perceptron algorithm,” *Machine learning*, vol. 37, no. 3, pp. 277–296, 1999.

[236] I. J. Good, “Probability and the weighing of evidence,” *Philosophy*, vol. 26, no. 97, 1950.

[237] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[238] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv preprint arXiv:1511.07289*, 2015.

Appendix A

Glossary

Micro-expression (ME)

Micro-expression are defined as involuntary, low intensity movements of facial muscles which occur in less than 0.5s.

Macro-expression (MaE)

Macro-expressions are defined as high intensity movements of facial muscles which usually lasts between 0.5 to 4s. It is also known as regular facial expressions.

Onset/Apex/Offset Frames

These frames are referred to the image sequence where facial expression (either ME or MaE) starts (onset), peaks in intensity (apex) and stops (offset).

Facial Expression Interval

This is defined as the total duration of facial expression (measured in seconds or milliseconds). It is the temporal difference of the onset and offset of the expression.

Micro-Expression Grand Challenge (MEGC)

This is the largest annual ME challenge for participants to compete and create the best model for ME related research (such as ME recognition or spotting). The criterion to measure the performance of the model are *precision*, *recall* and *f1-score* based on IoU method.

Intersect of Union (IoU) Method

A common criteria used in ME spotting that calculates the overlapping between the ground truth and prediction interval.

Facial Action Unit (AU)

Facial action units are individual facial movements based on Facial Action Coding System (FACS), a system that describes visible facial movements.

Appendix B

Fully End-to-End Approach on Composite Dataset

Our original ME and MaE spotting method is relies on OpenFace to perform face crop on the input image sequences. This approach takes this further by attempting to remove any dependence on external pre-processing and create an end-to-end deep learning pipeline for ME and MaE spotting. This pipeline lays a foundation for future spotting methods.

The main highlights are :

- Our approach is an end-to-end deep learning ME and MaE spotting method trained from scratch using long video datasets without relying on any other methods/software.
- Our approach shows MaE detection across two dataset with different frame rate and settings trained using a composite dataset.
- We found that the use of small amount of filter which focuses on extracting simple features is suitable and sufficient for spotting ME and MaE.
- Larger kernel size in earlier layers has a larger field of view which helps in capturing difference between frames.

Pipeline Comparison to Original Method

The frame input pipeline is identical to our original ME and MaE spotting approach which

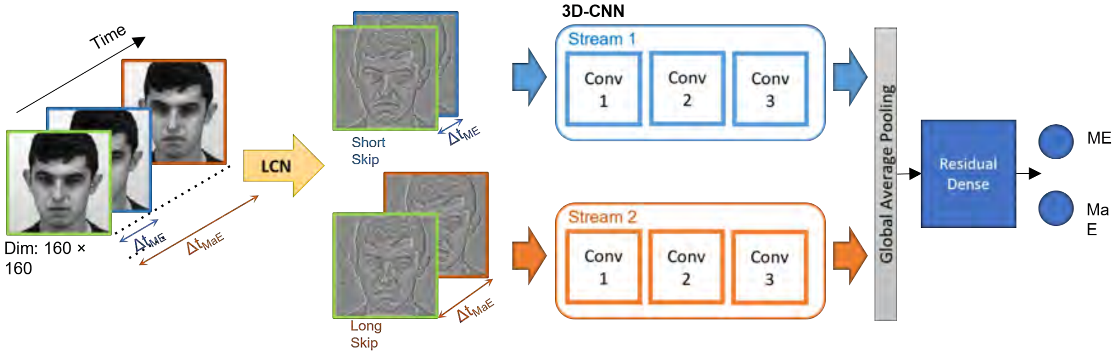


Figure B.1: The network is similar with previous method. The temporal oriented frame skip is identical. The main difference is each convolutional block consisted of 2 × (regular convolution, batch normalisation and dropout) and downsample using maxpooling. The input images have higher resolution of 160 × 160 pixels and face crop was not applied. The number of filter used in each convolution is lower (2, 2, 4, 4, 8, 8 for each conv layer in sequence). The kernel size for each convolution is different (11, 9, 7, 5, 3, 3 for each conv layer in sequence). The residual dense layer possesses the skip connections that shares weights. Two dense nodes were used at the end to resemble the presence of ME and MaE.

encodes temporal information using depth dimension of 3D-CNN. The network architecture is also a two-stream CNN with shared weights.

There are a few key differences in this pipelines. The input image sequences have a higher resolution of 160×160 pixels. There are more convolutional layers in this network (increased to 6). The number of filters used in each layers are lower. The kernel size for shallow layers are larger and become smaller as network depth. Regular 3D convolution is used and maxpooling is used for downsampling. The pipeline and detailed information are described in Figure B.1.

The training set for this experiment is consisted of a composite dataset which is combination of SAMM-LV and CAS(ME)^2. One of the subject from each dataset is used as evaluation set. Similar with previous method, randomised frame skip is used during training while k-frame skip is used in the test set (same as in Table 5.5).

Results and Discussion

The evaluation on one participant from SAMM-LV and CAS(ME)^2 are conducted. The raw output and ground truth comparison are shown in Figure B.2. It is noted that this is a raw output without any post-processing that shows our network has certain confidence on the intervals detected. However, a more suitable evaluation or post-processing might be needed as there are a lots of fluctuations especially near the onset and offset of the ground truth.

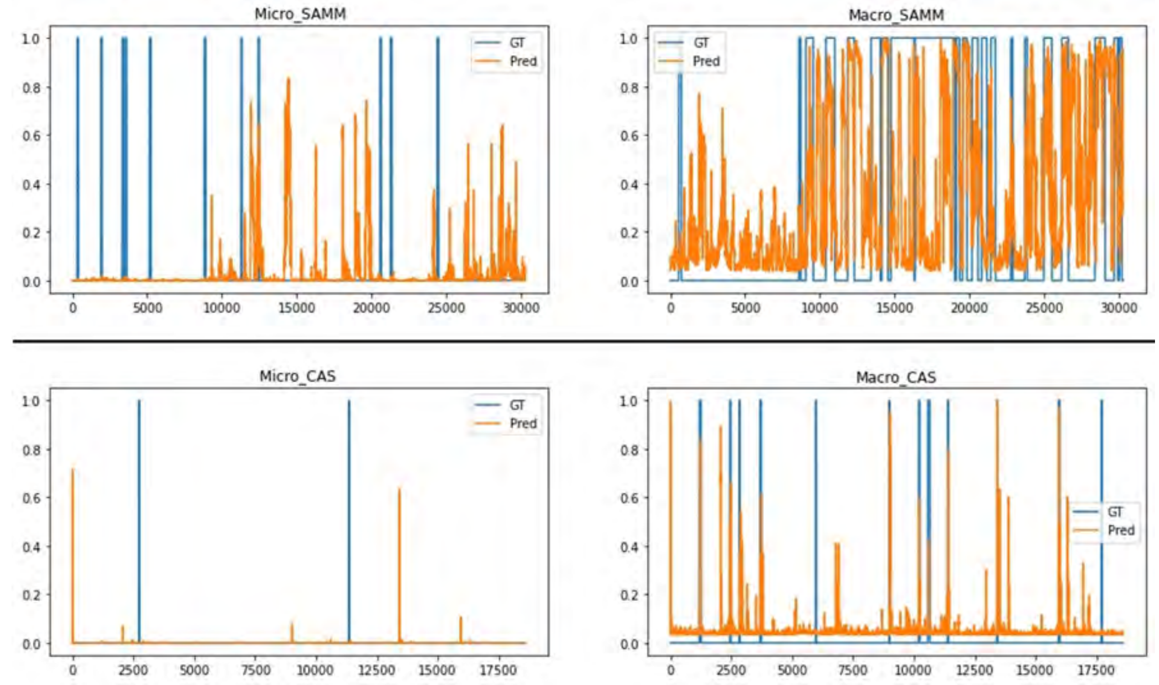


Figure B.2: ME and MaE spotting pipeline in fully end-to-end manner. These are the raw output of the model compared with unseen data from two different dataset. This model is trained using combined of two dataset of different frame rate (SAMM-LV and CAS(ME)^2). This model is able to capture MaE but is not sensitive to ME.

Using the post-processing of previous method, the model trained on composite dataset is evaluated with unseen dataset. The evaluation results using IoU on unseen dataset in MEGC2022 which uses both SAMM Challenge and CAS(ME)^3 only has detection on MaE. For SAMM Challenge, the F1-score is 0.0952; for CAS(ME)^3, the F1-score is 0.0816. The post-processing step of previous method might not be suitable or compatible with this method which might explained the drop in performance. However, visually the raw output shows promising results of spotting MaE.

Optical flow which measures the apparent motion of object functions similarly to the low number of filters used in this method. Facial movements can be thought of as difference between frames. Motion capture algorithm tracks the difference between frames. Using larger kernel size, change of pixels is easier to be captured. Low number of filters for each convolutional layers are sufficient as movements can be thought of as frame differences and simple features is suitable to detect them.

In this method, major facial features are detected which is beneficial in movement detection. However, detailed features is ignored most of the time which can be a reason on why no ME is detected using this pipeline. The guided backpropagation of each convolutional layers are visualised and confirmed the claim in which the major facial features are used in the decision making of the network as shown in Figure B.3.

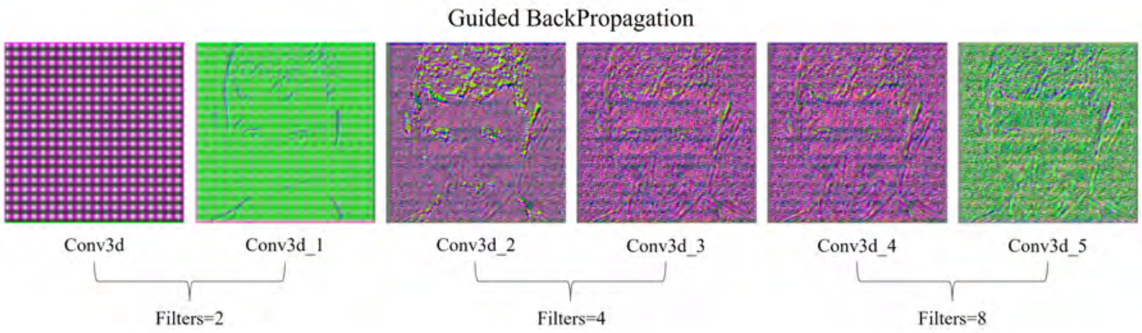


Figure B.3: Guided backpropagation on one subject from SAMM-LV dataset. The network detects and spots movement based on major facial features.

Appendix C

Additional Literature Review

C.1 Face Detection and Facial Landmark Detection

Viola-Jones algorithm [229] is used in the first real time face detector using Haar features (in Section 3.3). The method performs fast computation using intermediate representation using summed area table called "integral images" across all selected Haar features.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{C.1}$$

where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. Calculated using the following equation:

$$s(x, y) = s(x, y - 1) + i(x, y) \tag{C.2}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{C.3}$$

where s(x, y) is the sum of cumulative row, s(x,-1) = 0, and ii (-1, y) = 0.

Deng et al. [230] conduct facial landmark localisation by using a deep convolutional neural network. The author replaced the max pooling by depth-wise convolution in the CNN for better localisation performance. A response map for each facial points was defined as probability of presence likelihood and the model was trained using KL divergence loss. This algorithm takes the response maps of enhanced CNN and applies auto-encoder model to the global shape vector. The experiment was trained on the 300-W dataset [231].

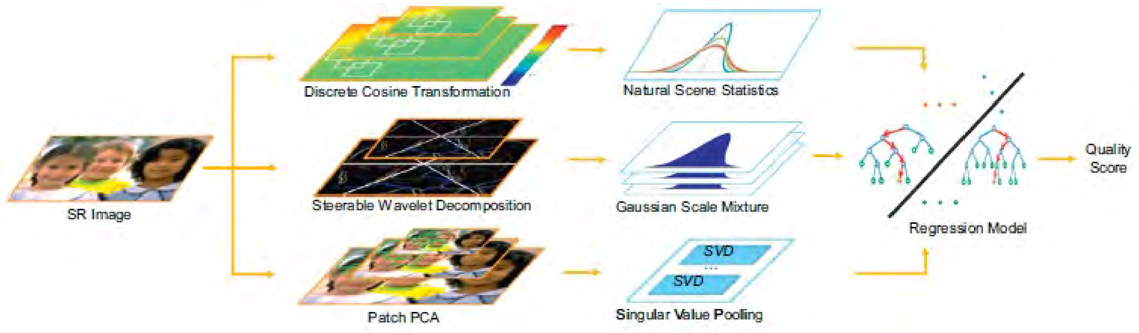


Figure C.1: Main procedures of Ma's score, a no-reference metric. Image originated by Ma et al. [97]

Bian et al. [232] shows a facial landmark detection method which involves self-adaptation model which processed to locate the eyes and mouth where the initialisation model will adapt to the real location. For further improvement, a multi-task feature learning was used which improve the generalisation performance.

C.2 Image Quality Analysis

Ma's score [97] introduces a no-reference metric learned from visual perceptual scores. This metric contains 3 types of low-level spatial and frequency domain based statistical features. It is used mainly to quantify artifacts produced in super-resolution tasks. The overall process is shown in Figure C.1. Local frequency features are transformed discrete cosine transform (DCT) to quantify high-frequency artifacts. This functions to obtain the DCT coefficient to fit the generalised Gaussian distribution. Global frequency features are extracted using Gaussian scale mixture (GSM) model that describe the marginal and joint statistics of natural images using a set of neigbouring wavelet bands. For spatial features, this method uses principal component analysis (PCA) to describe spatial discontinuity. A two-stage regression model is used to model local frequency, global frequency and spatial discontinuity with three independent regression forests.

Fowlkes-Mallows Index [233] (also known as G-measure) is the geometric mean of the precision and recall. This index has a value range of between 0 and 1. A high value implies a good similarity between two clusters. The mathematical equation for this index is as below:

$$F\text{-}M = \sqrt{precision \cdot recall} = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \tag{C.4}$$

C.3 Classifiers in Supervised Learning

Machine learning especially in supervised learning uses classifiers to assign a class label to data input. There are several classifier of supervised learning which are decision tree, perceptron, naive Bayes, k-nearest neighbour, support machine vector and artificial neural network.

Decision Tree is a flow diagram of all possible outcomes for certain choices. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume [234]. A decision tree normally starts with one node that branches out into possible outcomes. Each of those outcomes can further branch out into other possibilities. The common variations are the classification tree that is used in categorisation tasks and the regression tree that predicts continuous values.

Perceptron is inspired and modelled after the biological neuron. It is a binary classifier which decides whether an input represented by a vector of numbers, belongs to a certain class. The perceptron algorithm (a group of single-layer perceptrons) is used to learn from a batch of training samples by running the algorithm repeatedly until it finds a prediction vector which is correct on the training set. This vector is then applied for predicting the labels on the test set [235]. This algorithm is one of the earlier versions of the artificial neural network.

Naive Bayes Networks are very simple Bayesian networks which are composed of directed graphs with only one parent (representing the unobserved node) and several children (corresponding to observed nodes) with a strong assumption of independence among child nodes in the context of their parent [236]. Due to its simplicity, it requires fewer data to obtain good results in most cases.

K-Nearest Neighbour (KNN) is an instance-based learning. Instance-based learning algorithms compare new problem instances with only instances seen in training by delaying the generalization process until classification is performed. KNN assumes instances with similar properties within a dataset will generally exist in close proximity to each other [237]. KNN calculates the distance between test data and all the training points to predict the correct class of the test data. It selects the K number of points closest to the test data. A major weakness of KNN is the basic ”majority voting” classification that occurs in datasets with skewed class distribution. The more frequent class will often dominate the predictions due to their larger number.

Support Vector Machine is a supervised learning model primarily for classification and regression. The simplest form of this model creates a hyperplane that separates two data classes. It maximises the margin by creating the largest possible distance between the separating hyperplane [234].

Artificial Neural Networks is not technically an algorithm, it is a collection of algorithms working together. It is designed to mimic the problem-solving process of a human brain. A few examples are convolutional neural network (CNN), long short-term memory (LSTM) network and transformer (attention-based model).

For this thesis, artificial neural networks are used primarily because it has the potential to improve with more data, has the ability to execute feature engineering on their own and can perform complex tasks that linear program cannot. In the real world, problems are often complex and challenging. Artificial neural networks offer a solution to address these questions.

C.4 Activation Function

The activation function in artificial neural networks decides whether each neuron is active. Non-linear activation is commonly used as it produces a complex mapping between the network input and output by introducing a non-linearity relationship between each network layer. For machine learning, commonly differentiable activation is used as its gradient can be calculated which allows backpropagation. The notation used in this section are x represents input features, f represents activation function and N represents the number of input. The visualisations of the activation functions are shown in Figure C.2.

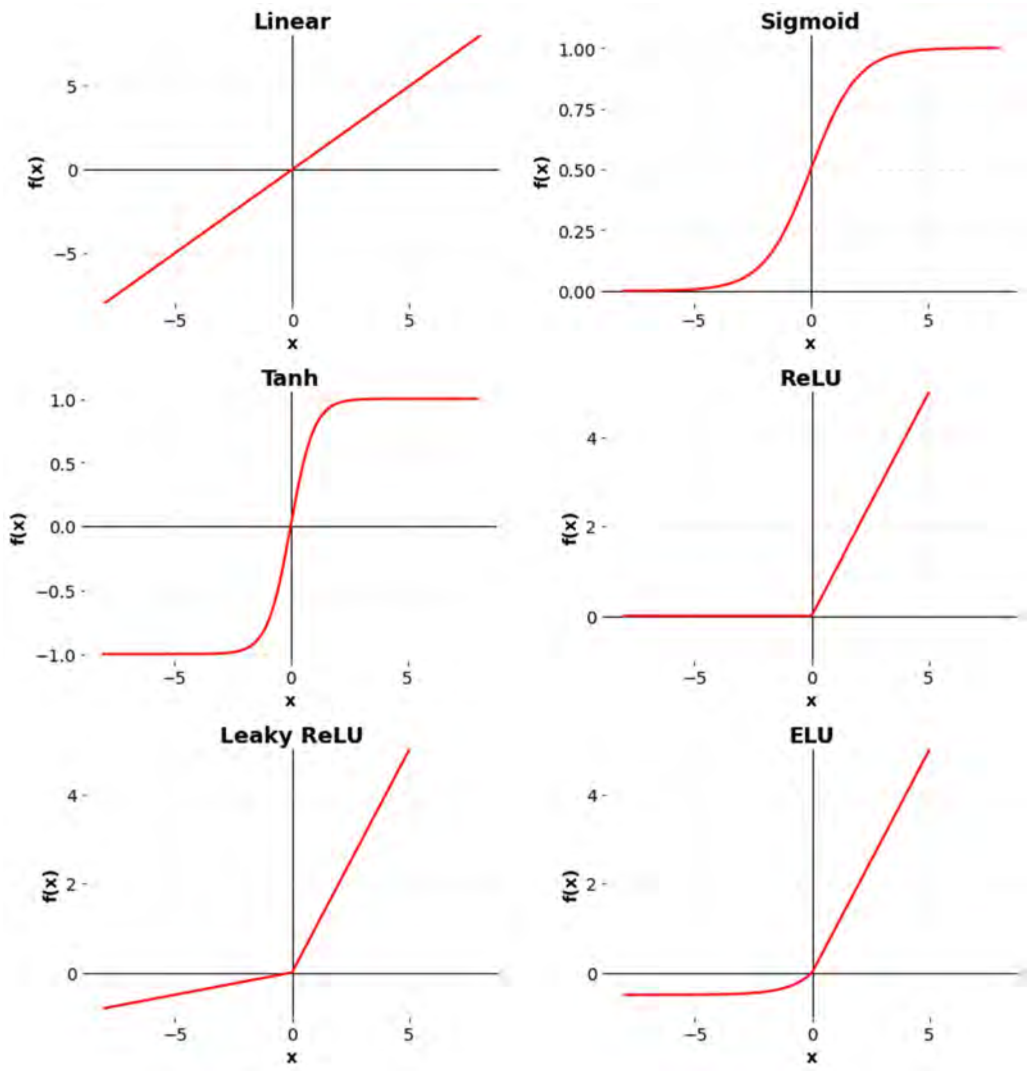


Figure C.2: Activation functions visualised where x is the input, $f(x)$ is the output of the activation function.

Linear activation is a simple linear function which has output identical to the input. This function can be used during the reshaping of hidden layers. However, it is not useful to create non-linearity in an artificial neural network.

Sigmoid activation is a non-linear function that normalises input to between 0 and 1. This function attempts to address exploding gradient problem by converting input of large intensity to a range of 0 and 1. It is originally used in an early artificial neural network as it mimics the activation of biological neurons. A common issue with this activation is the output saturates when there is a large positive or negative number causing the gradient to become almost zero.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{C.5}$$

Hyperbolic Tangent (Tanh) is similar to sigmoid activation with the difference being this activation normalises input to between -1 and 1. The main advantage is that the gradient obtained is four times greater than the sigmoid counterpart. The disadvantages are similar to the sigmoid function where the activation saturates on large numbers.

$$f(x) = tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{C.6}$$

Normalised exponential function (softmax) activation normalises the output of each class between 0 and 1 and performs division based on their sum. It is a smooth approximation to the arguments of the maxima (argmax) in which some function values are maximised. This activation is commonly used in the output layer of a neural network in multiclass classification.

$$f(x) = \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}} \tag{C.7}$$

Rectified Linear Unit (ReLU) has a linear activation for the positive input while remaining zero when input is zero or negative. This activation is identical to a half-wave rectifier. It is the most widely used activation as it is computationally efficient and has better gradient propagation. This makes the network converges quicker. However, this activation suffers from the "Dying ReLU problem" that occurs when inputs approach zero or negative. This causes the gradient to become zero and the weights to stop updating.

$$f(x) = max(0, x) \tag{C.8}$$

Leaky ReLU is based on ReLU with small negative values when the input is negative. This activation primarily addresses the "Dying ReLU problem". By updating a small negative gradient in the negative region, the weights will still update which theoretically prevents the activation from producing the same output.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases} \tag{C.9}$$

Parametric ReLU is also based on ReLU with negative values scaled using leaking coefficient, "*a*" when the input is negative. The "*a*" parameter is learnt by the network on its own using gradient descent. This activation also addresses the "Dying ReLU problem" by including a negative gradient.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ ax & \text{otherwise} \end{cases} \tag{C.10}$$

Exponential Linear Unit (ELU) [238] has identical output as ReLU when the input is positive. For negative input, the mean of activation is pushed closer to zero. Pushing the mean activation to zero enables faster learning as the gradient is said to be similar to the natural gradient. According to the author, ELU has a saturation in the negative region which makes the activation more stable and robust in learning representation.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha \cdot (e^x - 1) & \text{otherwise} \end{cases} \tag{C.11}$$

C.5 Loss Function

A loss function is defined as the function that describes the distance between the estimated and true values of a problem (commonly classification or regression problem). The larger the loss, the further the distance between the estimated and true values. Machine learning algorithms attempt to minimise this function to optimise its performance. The notations used in this section are x is the ground truth, y is the predictions, n is the total number of predictions made and L is the loss.

Mean Absolute Error (MAE) takes the average of absolute differences between the ground truths and predictions. This loss measures the magnitude of errors without the directions. All the errors are on the same linear scale. It is also known as L1 loss which represents

Least Absolute Deviations. The range is from zero to infinity.

$$L(y) = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{C.12}$$

Mean Squared Error (MSE) takes the average of the square differences between the ground truths and predictions. This loss is sensitive to outliers and will perform worse in datasets with outliers. It is also known as L2 loss which represents Least Square Errors. Similar to MAE, the range is from zero to infinity.

$$L(y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \tag{C.13}$$

Cross-entropy loss (log loss) measures the classification model with output between 0 and 1. The main advantage of this function is the learning rate is fast when the difference is large and it slows down when the difference is small as it uses negative logarithmic difference. The loss increases exponentially when prediction probability is low and decreases to near zero for correct predictions.

$$L(y) = -\sum_{i=1}^{n} x_i \cdot log(y_i) \tag{C.14}$$

Binary cross-entropy loss is a combination of Sigmoid activation (explained in Section C.4) followed by cross-entropy loss. It is commonly used in boolean classification tasks with two classes that are “0” and “1”.

$$L(y) = -(xlog(y) + (1-x)log(1-y)) \tag{C.15}$$

Hinge loss is a loss function which maximises margin objective. This loss increases linearly during misclassification and also penalises correct classification within a certain margin (when $|y| < 1$). By penalising weakly correct classifications, the class difference is made more distinct in which a higher activation is needed for the prediction to be correctly classified. This creates increases the margin or boundary between different classes. The output range is between -1 to 1 and the “*tanh*” activation is commonly used.

$$L(y) = max(0, 1 - x \cdot y) \tag{C.16}$$

C.6 Optimiser

An optimiser in machine learning is an algorithm that maps the input to the output. This involves changing the attributes of the neural network during training such as weights, momentum and learning rate. This optimisation makes the network better in reducing loss and improving model performance.

Gradient Descent is a first-order optimization algorithm. It uses the first-order derivative of a loss function. It calculates the direction where the weights should be changed for the function to reach a minimum. This is done using backpropagation. This function is computationally inexpensive and extremely simple to be implemented. However, this algorithm is very susceptible to being trapped in local minima and requires large memory for gradient calculation across the whole dataset.

Stochastic Gradient Descent (SGD) is based on gradient descent with more frequent gradient updates. It selects batches of data randomly (stochastic means random) and computes gradient descent. It optimises the objective function iteratively using suitable smoothness properties. The frequent updates will result in faster model convergence. It also has less memory requirement compared to gradient descent.

AdaGrad is an optimiser which fine-tunes the learning rates at each time step. The learning rate depends on parameter differences. It is a type second-order optimisation algorithm. It uses the second-order derivative of the loss function. AdaGrad considers all previous gradients. The drawback of this optimiser is it may decrease the learning rate too aggressively which interrupt the learning process completely.

AdaDelta is an extension of AdaGrad with a decaying learning rate removed and instead limits the previously accumulated gradients within certain windows. It uses an exponential moving average instead of the sum of all gradients. This reduces the tendency of aggressive learning rate decrement by AdaGrad.

Root Mean Squared Propagation (RMSProp) is a combination of gradient descent and AdaGrad that utilises decaying average partial gradients in the adaptation. This algorithm takes into account recent gradients more than previous gradients. The algorithm accelerates the optimisation process by reducing the number of evaluation functions used.

Adaptive Moment Estimation (Adam) works by adaptive optimisation of both the first and second order of momentum. It is a stochastic gradient descent method. It calculates the exponential moving average of gradient (first moment) and square gradient (second moment) of the loss function. Parameters β_1 and β_2 control the decay rate of the moving averages. The first and second moments of the loss function are calculated. This optimiser is one of the state-of-the-art commonly used in machine learning.