Noname manuscript No.
(will be inserted by the editor)

Predicting Humans Future Motion Trajectories in Video Streams using Generative Adversarial Network

Muhammad Ahmed Hassan, Muhammad Usman Ghani Khan, Razi Iqbal, Omer Riaz, Ali Kashif Bashir, Usman Tariq

Received: date / Accepted: date

Abstract Understanding the behavior of human motion in social environments is important for various domains of a smart city, e.g, smart transportation, automatic navigation of service robots, efficient navigation of autonomous cars and surveillance systems. Examining past trajectories or environmental factors alone are not enough to address this problem. We propose a novel methodology to predict future motion trajectories of humans based on past attitude of individuals, crowd attitude and environmental context. Many researchers have proposed different techniques based

Muhammad Ahmed Hassan
Department of Computer Science, UET, Lahore, Pakistan
National Centre of Artificial Intelligence, KICS, UET Lahore
E-mail: ahmed.hasan@kics.edu.pk

Muhammad Usman Ghani Khan
Department of Computer Science, UET, Lahore, Pakistan
National Centre of Artificial Intelligence, KICS, UET Lahore, Pakistan
E-mail: usman.ghani@kics.edu.pk

Razi Iqbal
Al-Khawarizmi Institute of Computer Science, UET, Lahore, Pakistan
E-mail: razi.iqbal@ieee.org

Omer Riaz
Islamia University, Bahawalpur, Pakistan
E-mail: omer.riaz@iub.edu.pk

Ali Kashif Bashir
Manchester Metropolitan University, Manchester, UK
E-mail: dr.alikashif.b@ieee.org

Usman Tariq
Prince Sattam bin Abdulaziz University, Saudi Arabia
E-mail: u.tariq@psau.edu.sa

on different features extraction and features fusion to predict the future motion trajectory. They used traditional machine learning algorithms like SVM,social forces, probabilistic models and LSTM to analyze the heuristic motion trajectories but they didn't consider the other environmental factors e.g relative positions of other humans present in environment and positions of objects present in environment which can affect the motion trajectories of humans. We intend to achieve this goal by employing Long Short Term Memory(LSTM) units to analyze motion histories, convolution neural networks to environmental facts e.g. human-human, human-object interaction and relative positioning of 80 different objects including pedestrians and generative adversarial networks(GANs) to predict possible future motion paths. Our proposed method achieved 70% lower Average Displacement Error(ADE) and 41% lower Final Displacement Error(FDE) in comparison to other state of the art techniques.

Keywords GAN · Future Motion Trajectories · LSTM · Object detection · Human re-identification · Path planning

1 Introduction

Humans possess an intuitive ability for navigation, which includes predicting the path trajectory of moving objects in the ecosystem for their better path planning. With the advancements in technology, navigation of automated guided vehicles (AGVs) and autonomous robots in static environment is not a big deal, but sharing the same ecosystem with humans for these machines is a challenging task. Moreover, the surveillance in this dynamic ecosystem is also a big problem for detection of suspicious activities [1, 2] and control over large crowds e.g. monitoring the suspicious areas with the help of autonomous and intelligent robots etc.
During the past few decades, terrorism has been the biggest menace faced by the research community. The surveillance of the suspicious objects, detection and the prediction of the unusual events using the autonomous robots can save the world from the large catastrophes "crush and stampede" on September, 2015 at Mina, Saudi Arabia, that resulted in the death of estimated 2000 Visitors, with suffocation and getting crushed [3]. Furthermore, on March, 2019 a mosque was attacked in New Zealand, which killed 51 and 49 got injured [4]. Another tragic incident took place at Strasbourg, where a man opened fire at civilians in a Christmas market, resulting in killing 5 and 11 got injured [5]. If these attacks can be obviated through human trajectory predictions many lives could survive [6, 7].
In light of all above, if the system can predict the future path trajectory of a moving or suspicious moving object, a number of such tragic incidents can be controlled and tackled. Moreover, the navigation of autonomous robots and Automatic Ground Vehicle's (AGV) in the ecosystem shared by humans will be more reliable and easier [8]. It will increase the perspectives of security and safety. Crime rate will drop overall as if the surveillance become easy, and it will become an easy task to catch a suspect or suspicious object by predicting in advance for his future motion trajectory [9, 10].
A community of researchers is continuously working to overcome this problem like Amir Sadeghian et al. [11] proposed a solution for path prediction for multiple agents interacting in a scene. They named it Sophie Framework based on Generative

Adversarial Network. Their solution comprises of three key modules. (a) feature extractor module, (b) attention module and (c) LSTM based GAN module. They trained their model on publicly used common datasets such as ETH, UCY, latest and ambiguous dataset of Stanford drone dataset.
Beomjoon Kim et al. [12] proposed a solution for socially adaptive path planning in a dynamic environment. Basically, they focused on wheelchair robot to move and predict their path in a crowded place. They employed it by generating human-like path trajectory. Basically, this framework is comprised of three modules (a) feature extraction module, (b) inverse enforcement learning, and (c) path planning module. All these above described works are more based on human-human interaction trajectories, but not focused on human's path planning trajectories depending upon human-human interaction. That is why their works are limited to an extent and cannot perform well in the environment where objects like vehicles and banners are placed on paths.
The motion behaviour of a human can be driven by many factors, like the actions made by surrounding agents may derive the target agent to move, or its own goal impulses the agent to move. So, both external and internal stimuli may force a target to change the position. The main issue with human motion detection is that most of the factors like geographical information,human-human and human-object interaction that makes a target to move are not directly observable; we have to infer it from complex and noisy perceptual cues like depth estimation to calculate the distance between the different objects and road segmentation for geographical information extraction. Moreover, for an intelligent system to be very accurate and effective in practice, motion prediction has to be very fast in real-time. In order to make the accurate motion positions, path trajectory plays an important role [13]. So, prediction of path trajectory for such a moving body even in a dynamic and congested ecosystem by analyzing the motion heuristics and environmental context is the focus of our research work. This research work utilizes the state of the art techniques, LSTM-GAN with feature fusion to achieve the lowest ADE and FDE. The main contributions of this research work are as follows:

- We incorporate the regional CNN(Convolution Neural Network) to localize the objects present in an environment which may affect the motion trajectories of humans.
- We introduce a pooling layer which is able to fetch the relative distance between the human-human and human-object instead of only human-human which is performed by previous research.
- According to best of our knowledge there is no existing dataset which contains the interaction of humans with local objects.So, we generate the data set from local environment which holds the information about the human-local-object interaction.

The paper is organized in the following manner. Section 2 explains the related state-of-the-art work. Section 3 describes the proposed technique, while Section 4 explains statistics and source of generated dataset. Section 5 provides about implementation details. Section 5.1 describes the evaluation measures and the results are discussed in section 6.Finally, in section 6.2 the proposed work is concluded and discussion about future directions is presented.

2 Related Work

The work on human motion trajectory prediction can be divided in two eras, traditional approaches and Deep Learning based techniques paradigms [14]. In the era of traditional techniques, the most famous and mostly used were social forces, multi hypothesis tracking, Guasian Process dynamic model and Modified Hausdorff Distance. Where as in the era of deep learning, SVM, semi supervised learning, unsupervised learning, LSTM and RNN are commonly used.

In the era of traditional techniques the behavior of human in crowd has been studied from macroscopic models using social forces. Helbing and Molnar [15] used the social forces in the crowd as an example of macroscopic model. They analyzed the motion of people in the crowded area and guided the person to achieve his goal. To enhance the work further Matthias Luber et al. [16] suggested a solution to track persons motion and predict his path. They also used social forces to do this. They performed their experiment in outdoor as well as the indoor environment, They combined social force model with MHT "Multi hypothesis tracking" based on the work of "Reid". The Social force model is used for predicting humans short term motion as where the person moves next.

The results of the previous works were not satisfactory enough for human trajectory prediction, In 2016, L. Ballan et al. [17] worked on the path prediction of pedestrian and cyclist. They analyzed the previous motion and velocity of pedestrian and cyclist to predict their future paths. The prediction also includes the future velocity of objects. They used UCLA-courtyard and Stanford-UAV dataset to train the model. Modified Hausdorff Distance(MHD) was used for evaluation. They achieved the loss of 10.32 on UCLA-courtyard dataset and 8.44 on Stanford-UAV dataset which is not satisfactory.

Extending the work further based upon mapping and MHD, P. Coscia et al. [18] created three different types of maps of environment and by using these maps they predicted the future motions of humans. The three maps are semantic map, heat map and desirability map. These maps contain the information about the environment and the objects present in the environment. The semantic map was actually a segmented image of environment where each object is localized, heat map contained the information about the moving object in the environment and desirability map contained the path information where the humans can move such as roads. They also used MHD for model evaluation and achieved 3.56 error.

With advancement of knowledge in traditional techniques, the introduction of SVM provided revolutionary changes. Recently it is used to solve many problems like activity recognition [[19], [20]], EEG signal classification [21] and fault detection [[22], [23]]. Satoru Satake et al. [24] proposed a solution for the robots to do "conversation with the people walking in the environment". They used the approach of "Predicting the walking behavior of people, choosing a target person, planning its approaching path, and then to make guess the intention of people to do conversation with predicting the future path of the people, By using SVM they classified 2 seconds of path trajectory in four of classes fast walking, Idle, wandering, stopping. with this proposed method robot was successful in 33 approaches out of 59 trials. The failure rate at each step in the proposed and simple approach is at unreachable 3%, 25%, at unaware 4% to 14%, at unsure 18% to 24% and at rejected 27% to 29% respectively.

The era of traditional techniques didn't achieve good results in term of loss and

accuracy. But after the birth of deep learning every problem boost its performance and accuracy. But deep learning is data hungry and require huge computational power because of its large number of parameters. but these days the parallel computing is possible by invention of GPUs and large datasets also not a big deal. In deep learning era, Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) based techniques are used to learn the temporal and sequence to sequence models. RNN is rich extended form of sequence models. It is used as feed-forward networks for sequence generation in speech recognition [[15], [25], [26]] and in many fields of natural language processing like machine translation [25], text summarization [27] and video summarization [28] [29]. It is also used in image and video captioning [[30], [31], [32], [33], [9]]. It gives very outstanding results over traditional techniques. J. Liu, et. al. [34] used RNN to get spatio-temporal information for action recognition. Many networks used LSTM and RNN to solve complex human-human and human-environment interactions [[35], [36], [37]. Alahi et al. [35] used social pooling layer to map the motion information of other people present in the environment. Lee et al. used encoder-decoder architecture based on RNN to predict the motion trajectories of human in a crowd. But they did not used the human-human interaction in crowded area. A. Alahi et. al predicted the future motion of people in the crowded scenes. For this they used the one LSTM for each person which analyze the past position of humans from time T_0 to T_{obs} and then predicted the motion of human for time grater then T_{obs}. The position of human at any time can be represented by spatial coordinates like at time T_j the position of human can be presented as (xtj, ytj). As they used one LSTM for each person it means they are not dependent on other human's motion in the environment. T. Fernando el. al [38] proposed the architecture to predict the future motion trajectories of human in the environment using LSTM encoder and decoder. The prediction is dependent on the past motion of human and also dependent on the distance between the humans. They added some dependency of human motion on other human behavior. They achieved better results then A. Alhai [35].
Proceeding the deep learning work further by the utilization of the latest trends, Kratarth Goel. et al. [39] presented a completely unsupervised framework to learn the similarities behind the human motion. Their model considers everything in it i.e. (humans, Bicyclist, Skateboarders, Drivers), etc. This is opposite to typical approaches which use "social forces". They use recurrent end to end convolutional architecture to guess where the target will move next. They use both "CNN" and "LSTM" that are recurrent network to successfully implement their model. Truly their model is not error prone but it outperforms the previous models on public datasets as well as a newly introduced dataset. They used Isengard and Hobbiton datasets to implement the model by dividing it in classes and achieved average displacement error (ADE) in between 8.9657 and -8.8521 which was not satisfactory enough.
After it Beomjoon Kim et al. [12] proposed a solution for socially adaptive path planning in a dynamic environment. Basically, for the wheelchair robot to move and plan their path in a crowded place. This is done by generating human-like path trajectory. They used verse reinforcement learning to do the job, on the time it was the latest framework developed to implement the machine learning. They use the gathered data and apply it on the three-test scenarios (a) Pedestrian walking towards the Robot (b) pedestrian walking horizontally to the Robot (c) and multiple pedestrians. They repeat each scenario ten times and then calculate

the “confidence interval” value which is 95 %. The confidence interval is the factor on which their result depends on.
According to best of our knowledge and literature survey the eras of traditional approaches and deep leaning focused either on previous motion or the environmental information like maps and the position of other pedestrians to predict the future motion behaviour of humans and crowd but no one used the fusion of both features as, the both features are very important and effect directly on the motion trajectory of human.

To overcome the problems from previous work, we propose a novel method which is able to perceive both features; environmental contexts and the previous motion trajectories for better and more real trajectory prediction.

3 Methodology

The proposed methodology consists of two main modules. The first one is generating the motion trajectories of humans and localize the objects from video stream in real environment coordinates. The second is predicting the future motion trajectories of humans.

3.1 Generation of Motion Trajectories

The generation of motion trajectories from video stream requires human and re-identification. Human detection provides the location of human in pixels which is converted to real environment coordinates as describe in section 3.1.1. Human re-identification is also required to determine the location of specific human in next frames of video so we can track the human in video stream to build proper motion trajectories. The human detection and re-identification techniques are described below:

3.1.1 Human Detection

Detection of pedestrian in social environment[1][2] and in crowded scenes[3] has been performed by many researchers. Recently, Faster-RCNN proposed by Ross Girshick[4] outperforms all previous methodologies in detection problem. So, we use Faster-RCNN model for human detection. The architecture of Faster-RCNN is described in figure 1.

Faster Regional Convolution Neural Network in short F-RCNN have four parts which play different role in the detection of objects. The different parts of F-RCNN are explained below:

Features Extraction The first part of F-RCNN is featured extraction. Deep features are used for this problem. VGG16 [40] architecture trained on ImageNet [41] dataset is used to extract the features.

VGG16 network has 15 convolution layers and three fully connected layers and one softmax layer at the end of the network. First, two convolution layers have

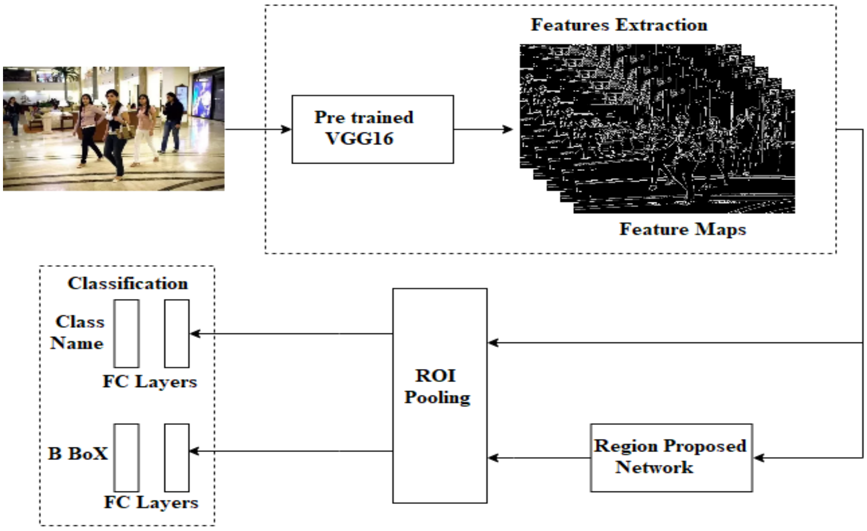

Fig. 1: Faster R-CNN for Human and Object Detection

64 filters, 3rd and 4th convolution layers have 128 filters, 5th and 6th convolution layers have 256 filters and last five convolution layers have 512 filters. Three fully connected layers have 4096,4096 and 2622 neurons respectively. Max pooling of 2x2 is applied after every block.

We extract the features after the last convolution layers. 512 feature maps are extracted which are further used by Region Proposed Network (RPN).

Region Proposed Network RPN accepts the features maps produced by the VGG16 model and proposes different regions on the features map where the humans can be present.

ROI Pooling The regions proposed by the RPN have arbitrary sizes. To make them uniform ROI pooling layer is used which accepts the regions proposed by the RPN and make them uniform. The main objective of ROI pooling is to provide the fixed length features to fully connected layers. The functionality of ROI pooling is explained in Figure 2.

Classification The fixed length of features produced by the ROI pooling is further used by the fully connected layers. There are two pipelines of fully connected layers. One to predict the region of interest for object and the second for predicting class label of object present in ROI as shown in the classification section in Figure 1.

3.1.2 Human Re-identification

Spatio-temporal features are required for human re-identification. Recently, recurrent neural networks have shown very excellent results. It has been used in many different problems like prediction and recognition of human's dynamics[5] and

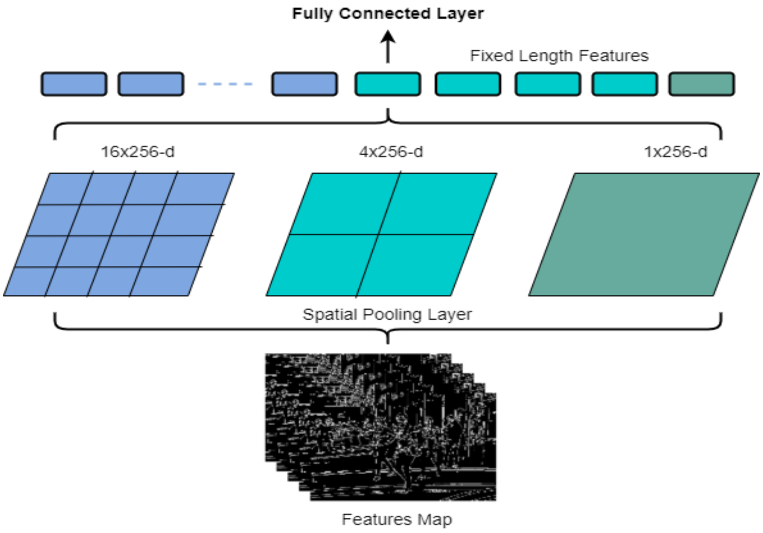

Fig. 2: ROI pooling module of Faster R-CNN

human action recognition[6] where temporal features plays vital role in learning of model. A Recurrent Faster-RCNN model is used by introducing LSTM units after detection. The LSTM units get location of human as well as the hidden features from Faster-RCNN and predict the locations of humans in next frames. Procedure of human tracking in the video stream is explained in Figure 3.

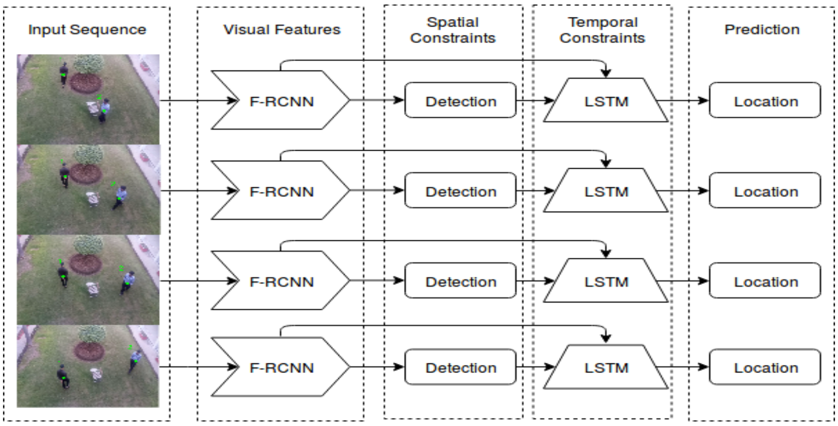

Fig. 3: Architecture of Recurrent Faster-RCNN

The spatial hidden features from Faster-RCNN of frames are passed to LSTM unit and it predicts the location of human.

Here F_t is features extracted by Faster-RCNN at time t and E_{t-1} is embedding of human's location at time $t-1$ which is used as temporal information. W represents the weights and b represents the biases. The location of human contains value of x-axis and y-axis. The embedding is generated by using equation 1.

$$e_i^t = \phi(x_i^t, y_i^t, W_{ee}) \tag{1}$$

Where x_i^t and y_i^t is values of x-axis and y-axis of human i at time t, and w is weight matrix. The results of human detection and re-identification is shown in figure 4.

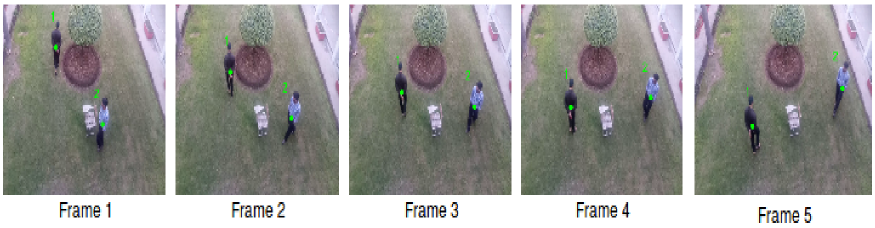
Frame 1
Frame 2
Frame 3
Frame 4
Frame 5

Fig. 4: Sample results of Human Re-identification

The motion trajectories are the positions of humans with respect to time. After detection and re-identification of human's location in frames, we get the position of the human in pixels coordinates. But actually the human moves in real coordinates not in pixels coordinates. We map the pixels coordinate to real-world coordinates(meters). To convert distance from pixels to meters, we follow these steps:

- Calculate the dimension of camera stream (width and height)
- Measure the area in real world covered by the camera (manually)
- Calculate the number of meters covered by a pixel using equation 2

$$MPP = \frac{M_{total}}{P_{total}} \tag{2}$$

The results after conversion in coordinates are shown in fig 5.

3.1.3 Object Detection

The motion of human is also affected by the position of objects in the environment. The humans plan their path by analyzing the motion of other humans and the position of objects. To predict the future motion of human, the position of the objects is very important. To detect the objects in images/video streams we used regional convolution neural network named Faster-RCNN. The faster R-CNN was trained on MS-COCO [42] data set as it has more outdoor object classes which is more beneficial for our research work. It contains 80 classes of objects. We used pre-trained weights learned on MS-COCO dataset. The output we get from F-RCNN is the region of interest. But we need a point in spatial coordinate. We calculate the average of both axes and get a pivot point of an object using Eq. 3 and 4.

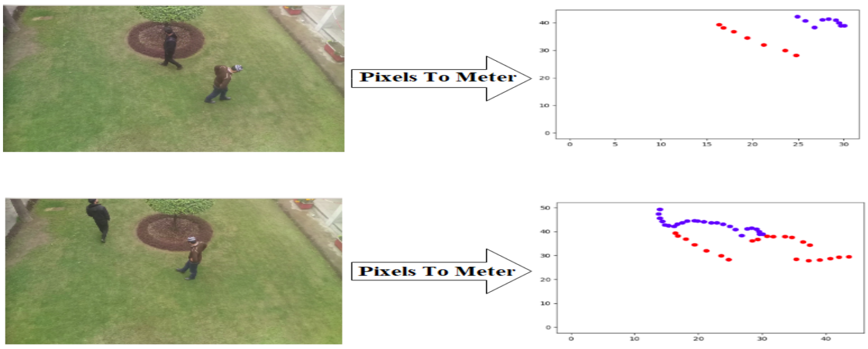

Fig. 5: Sample Results of Pixels to Meters Conversion

$$X = \frac{X_{max} + X_{min}}{2} \tag{3}$$

$$Y = \frac{Y_{max} + Y_{min}}{2} \tag{4}$$

The output results of this module is shown in figure 6. We assign a unique ID to every object to make their tracking easy.

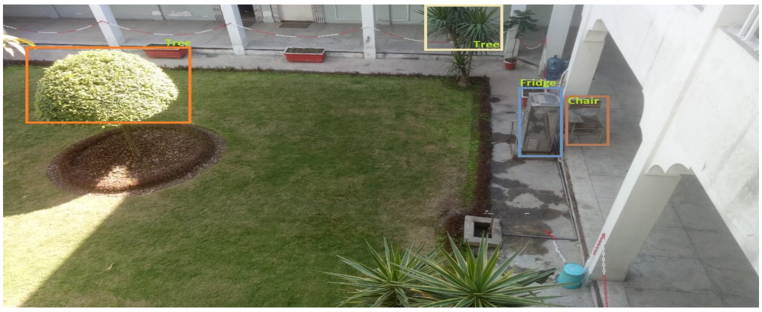

Fig. 6: Sample Results of Object Detection

3.2 Prediction of future motion trajectories

In first part of methodology, the motion trajectories and the location of objects in environment is generated. Now the seconds part focusses on the prediction of future motion trajectories of a human by using information extracted from part 3.1. To predict the future motion trajectories of human, we proposed an encoder decoder based Generative Adversarial Network(GAN). As the encoder and decoders are based on LSTM, we called it LSTM GAN. The functionality of one LSTM unit is describes below:

3.2.1 LSTM GAN

LSTM GAN is Generative Adversarial Network containing two neural networks based on LSTM [43]. These two neural networks are trained to oppose each other. The first neural network is called generator which produces some meaning full information from some features or scratch data. The second part is called discriminator which verifies the information generated by generator either it is real or fake as shown in Figure 7.

The discriminator already has real information available and uses this while verifying the information generated by the generator. The proposed LSTM GAN has four main modules.

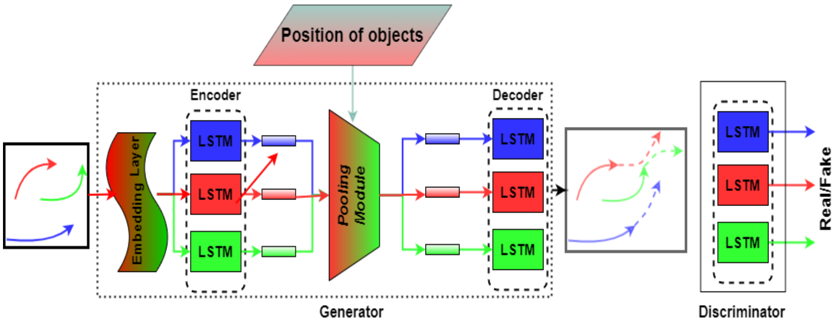

Fig. 7: Architecture of LSTM GAN
Non-dotted arrow represents the observed motion trajectory and dotted arrow represents the predicted motion trajectory

- Embedding layer
- Generator
- Pooling Module
- Discriminator

The learning of GAN is very similar to the min-max problem. The objective function of learning can be written as equation 5.

$$min_G max_D V(G, D) = E_{x=pdata(x))}[logD(x)] + E_{z=p(z)}[log(1 - D(G(z))] \quad (5)$$

Embedding layer: The input trajectories are of varying length but our network needs a fixed length inputs. We used MLP to generate the embedding of motion trajectories. These embeddings are of fixed length. These embeddings are used to be fed in the generators encoder.

Generator: The generator is a core part of any GAN, it is also called encoder. It is the only part which is used at the time of testing and deployment and the remaining parts are just used while learning. The generator G takes variable z as input and outputs sample $G(z)$. It tries to fool discriminator by producing real looking data. The generator of proposed GAN is built with layers of LSTM units. The generator has two parts encoder and decoder.

Encoder: The encoder takes embeddings of past motion trajectories of a human for 3.2 seconds generated by the embedding layer and predicts the embeddings of future motion trajectories for 3.2 seconds which are further decoded by the decoder. In other words, we take 8 steps, one step after every 0.4 seconds of past motion and predict the 8 steps in future. Encoder at time t introduces the following recurrence:

$$e_i^t = \phi(x_i^t, y_i^t, W_{ee}) \tag{6}$$

$$h_{ei}^t = LSTM(h_{ei}^{t-1}, e_i^t, W_{encoder}) \tag{7}$$

where w_{ee} and W encoder represents embedding weights and encoder weights respectively. $\phi()$ is embedding function with ReLU non-linearity.

$$ReLU(x) = max(0, x) \tag{8}$$

Decoder: The second part of the generator takes the embeddings predicted by the encoder and decodes it to real world motion trajectories. The functionality of the decoder can be expressed through the following equations:

$$e_i^t = \phi(x_i^{t-1}, y_i^{t-1}, W_{ed}) \tag{9}$$

$$P_i = PM(h_{d1}^{t-1}, \dots, h_{dn}^t) \tag{10}$$

$$h_{di}^t = LSTM(\gamma(P_i, h_{di}^{t-1}), e_i^t, W_{decoder}) \tag{11}$$

$$(\hat{x_i^t}, \hat{y_i^t}) = \gamma(h_{di}^t) \tag{12}$$

where w_{ed} and W decoder represents embedding weights and decoder weights, respectively. PM is pooling module and $\gamma()$ is an MLP.

3.2.2 Pooling Module

Pooling layer is used to incorporate the relative positions of one human to other humans and the objects into network learning as shown in figure 8.

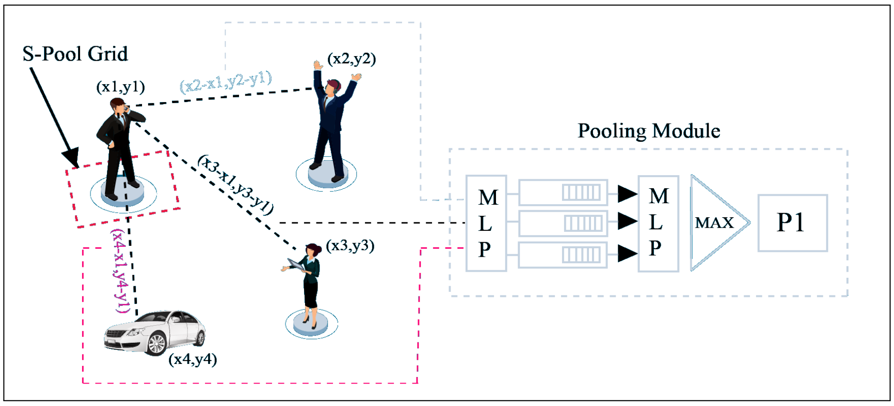

Fig. 8: Calculation of distance between human-human and human-objects

3.2.3 Discriminator

Discriminator contains its own encoder which accepts the trajectories generated by generators decoder and classifies them as real or fake by using its knowledge provided by the ground truth. The encoder consists of LSTM units and applies MLP as encoders last layer so we can get the score of real and fake. This score will be used to calculate the loss for optimization purpose.

4 Dataset Generation

This research work also makes a contribution towards generation of robust dataset for human motion trajectories. According to best of our knowledge, there is no such dataset which gives us the trajectories of humans in social environment as well as environmental contexts like position of objects. The Agrim gupta [44] used different datasets such as univ,zara and eth. etc, which only contains the position of humans with respect to time in the environment. The generated dataset is recorded by videos in social environments where the objects are also present with humans. The main sources of videos are parks, shopping malls, footpaths, bus stands and different locations in educational institutes. Some sample images are shown in Figure 9.We used different cameras like CCTV installed for surveillance at public places and handy cameras to record the videos. The statistics of generated dataset is described in table 1. While experiments we observed 8 steps and predicted the future 8 steps of pedestrians. Each step is taken after 0.4 second. So, our one sample utilizes 6.4 seconds of video. And in 60 minutes video we get 562 samples. The batch size we used is 256 and from 60 minutes video we get approximately 2 batches.

After recording the videos, we extracted the human trajectories using human detection and re-identification with position of objects. Storing this information in a structure is very important so, we can provide it to model easily while training. So, we stored the human trajectories in the CSV files. The stored data have 4 attributes:

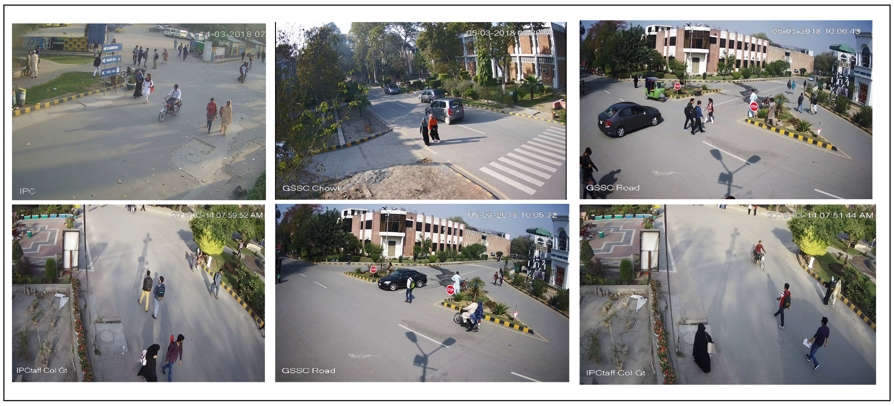

Fig. 9: Sample Images of generated Dataset

Place	**Duration of Video per location**	**Number of places**	**Total Video Duration**
Parks	60 minutes	2	120 minutes
Shopping malls	60 minutes	4	240 minutes
footpaths	60 minutes	2	120 minutes
Bus Stands	60 minutes	5	300 minutes
Educational Institutes	60 minutes	10	600 minutes
Total Duration of video	1,380 minutes		

Table 1: Dataset Statistics

- Frame number
- Person/object ID
- Position of person/object in X-axis
- Position of person/object in the Y-axis

The summary of the stored data in the CSV files is shown in Table 2.

Frame ID	Person.Object ID	X-axis	Y-axis
780	1	8.46	3.59
790	1	9.57	3.79
800	1	10.67	3.99
800	2	13.64	5.8

Table 2: Format of generated Data

5 Implementation details

The implementation details is describes in Table 3.

The loss function used for error calculation while training is Mean Square Error (MSE). The mathematical expression explained in equation 13.

Parameters	**Value**
Operating System	Ubuntu 16.04
Frame Work	PyTorch
Language	Python 3.5
CPU	Core-i7 (7th Gen.)
RAM	16GB (DDR3)
GPU	1080 Ti (11 GB memory, 3584 cores)
Batch size	256
Epochs	600
Drop Out	Dynamic
Learning Rate	Dynamic
Loss Function	Mean Square Error (MSE) eq. 13
Optimizer	SGD

Table 3: Parameters and their values while implementation

$$d(p,t) \quad = \quad \sqrt{(p_1-t_1)^2+(p_2-t_2)^2+\cdots+(p_i-t_i)^2+\cdots+(p_n-t_n)^2} \tag{13}$$

Here p and t is predicted and actual labels of the mini batch provided while training, p_1 shows the result of the 1st sample predicted by the model and the t_1 shows the ground truth of 1st sample.

After the calculation of loss by discriminator, the parameters of the generator should be optimized using some optimizer function. As the training data was very large and it requires a huge amount of memory to load all data at once. The solution to this problem is to split the data into mini batches and optimize the parameters of the model. Stochastic gradient descent optimizer was used for model optimization using a mini batch. SGD optimizer updates parameters by using given data $x(i)$ and label $y(i)$ as explain in equation 14

$$\theta_n = \theta_o - \eta.\ \bigtriangledown_\theta\ J(\theta; x^{(i)}; y^{(i)}) \tag{14}$$

Here θ_n are new gradients, θ_o are old gradients, η is learning rate, J is optimization function and i is total size of data.

As described above we used mini-batches to optimize the parameters instead of one sample so, the optimization of parameters is performed using equation 15.

$$\theta = \theta - \eta.\ \bigtriangledown_\theta\ J(\theta; x^{(i:i+n)}; y^{(i:i+n)}) \tag{15}$$

Here n is size of mini batch. The gradients calculated by the data mini batch partially update the learning parameters of the model using the dynamic learning rate. The value of learning rate is between 0 and 1, 0 means the parameters are not updated at all and 1 means the parameters are completely updated. The value of the learning rate defines the learning performance of the model. A smaller value of learning rate makes the model more accurate but requires much time and the larger value of learning rate optimizes the model fast but it will not perform well at testing time. So, we used a large learning rate at the beginning of training and gradually decreased it along the epochs as shown in table 4.

Epoch	**Learning Rate**
0-100	0.1
101-200	0.01
201-300	0.001
301-400	0.0001
401-6000	0.00001

Table 4: Learning rate with epoch's range

5.1 Challenges

SGD optimizer navigates to global minimum loss by taking small step in the direction, where the loss decreases. But sometimes the SGD stucks into local minima because there is no next point where the loss reduces. So, it stucks in local minima and optimization stops as shown in Figure 10.

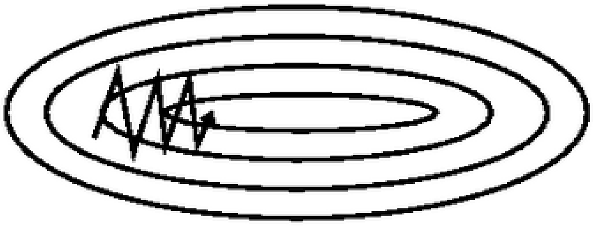

Fig. 10: Optimization without momentum

This problem is solved by momentum. The momentum accelerates the SGD to move in the desired direction as shown in Figure 11

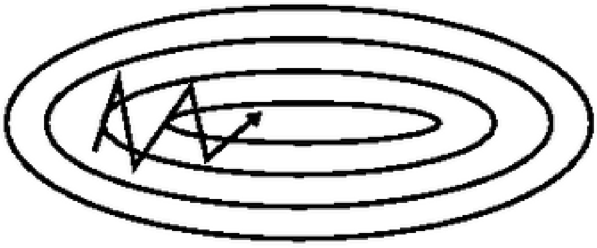

Fig. 11: Optimization with momentum

Momentum works by adding the fraction (lambda) of the last update vector in the currently updated vector as explained in equations 16 and 11.

$$v_t = \gamma v_{t-1} + \eta \bigtriangledown_{\theta} J(\theta) \tag{16}$$

Here γ is momentum which is multiplied with older gradients.

$$\theta_n = \theta_o - v_t \tag{17}$$

We used different values of lambda and got the best optimization results with value of 0.9.

6 Results and Discussion

The model was trained for 600 epochs. As show in figure 12 the training and testing loss reduces very fast till 500 epochs. After 500 epochs the loss is still reducing but very slowly and there is no overfitting so, we continue the training for next 100 epochs. After 600 epochs the loss graph becomes straight so, we stop the training and use trained weight at 600 epochs.

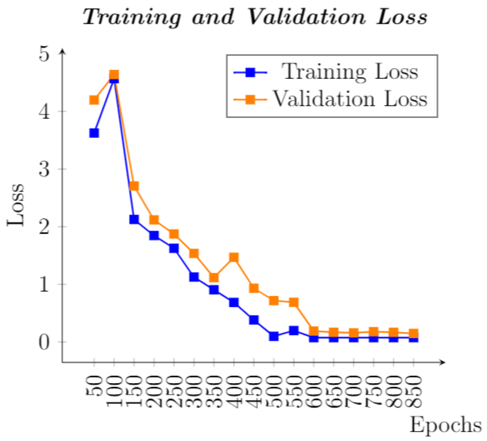

Fig. 12: Mean Squared Error

After every 50 epochs we evaluate the learning of the model by using two evaluation metrics, Average Displacement Error (ADE) and Final Displacement Error (FDE). To describe and measure the accuracy of the objects motion, we have to find and specify the exact object location based on the vector points. A vector is the quantity, which has the magnitude as well as the direction with respect to the other points in the space. In generic, we can say that we have to find the object position with respect to the time and ground truth reference frame to measure the accuracy of objects motion. To describe this, lets take the example of the earth. Earth is often taken as the reference frame and we describe the position of the specific object as it acts as the stationary with respect to the taken reference

point. If the particular object moves with respect to the reference object, like if the person moves towards the aeroplane, then it changes the position with respect to the under consideration object. This change in position with respect to the time is known as the displacement. The word displacement depicts that object has been displaced or been moved from its original position.
As mentioned earlier, displacement is the vector quantity that is specified by the magnitude as well as the direction and represented by the arrow pointing from the initial position to the final position. We have utilized two different approaches to analyse the qualitative measure of our proposed methodology. These approaches include final displacement error and average displacement error. We have discussed these approaches separately in the section described below.

6.1 Final Displacement Error (FDE)

The final displacement error is defined as the distance which is measured between the true final destination and the destination of the object which is predicted by the proposed architecture. We have utilized the equation 18 to calculate the final displacement error.

$$f_{disp} = \sqrt{\left(x_{t(8)} - x_{p(8)}\right)^2 + \left(y_{t(8)} - y_{p(8))}\right)^2} \tag{18}$$

From the equation 18, it is clearly shown that we have two points. These points contain the x and y axis value of both actual destination points and the final destination points. Here in equation 18 f_{disp} represents the final displacement error. Whereas, $x_{t(B)}$ and $y_{t(B)}$ represents the points of the actual destination of the object and the $x_{p(Nb)}$, $y_{p(B)}$ represents the predicted destination of the object. The final displacement is the difference between the actual object position and the predicted object position.

6.2 Average Displacement Error (ADE)

In average displacement error, we compute the displacement between the ground truth path and predicted path based on the selected intervals. Then we took the average of the displacement which we extracted for each step.

$$Avg_{disp} = \frac{1}{8}\sum_{i=1}^{8} \sqrt{\left(x_{t(i)} - x_{p(i)}\right)^2 + \left(y_{t(i)} - y_{p(i))}\right)^2} \tag{19}$$

From the equation 19 Avg_{disp} , represents the average displacement, for the fixed interval. In our case we have taken the eight different points on which we measure the displacement between the actual and the predicted paths of the specific object. Here, $x_{t(B)}$ and $y_{t(B)}$ represents the points of the actual destination of the object and the $x_{p(Nb)}$, $y_{p(B)}$ represents the predicted destination of the object. The average of the displacement of the eight different points represent the average displacement error.

The ADE starts from the 0.9865 and becomes straight after 600 epochs by achieving 0.2795 value and FDE starts from 1.1055 and decreases to 0.5635 after 600 epochs. The graphs of FDE and ADE is shown in figure 13 and 14 respectively.

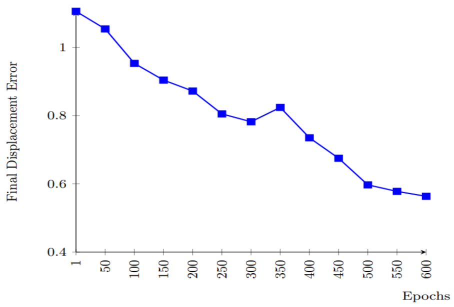

Fig. 13: Final Displacement Error

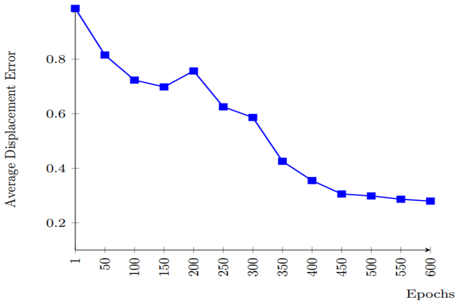

Fig. 14: Average Displacement Error

Some predicted humans trajectories by analyzing the past motion trajectories are shown in figure 15.The blue dotted line shows the observed motion trajectory, the red dotted line shows the motion trajectory predicted by our model and the yellow dotted line represents the ground truth motion trajectory.

Blues dotted line shows the observed motion trajectories or input of the model and the red dotted line shows the predicted motion trajectories and yellow line shows the actual motion trajectories of the human.

As shown in figure 15 the predicted trajectories are very close to actual trajectories but sometime is confused when the human performs some unusual motion. These unusual motions of human are depending on the geographical information like the turn on road and the end of path etc. As earlier mentioned in section 4 according to our best knowledge there is no such dataset with geographical context and human-human, human-object interaction. So, we evaluate the different techniques on our generated dataset with extra challenge of collision with objects. The comparison of

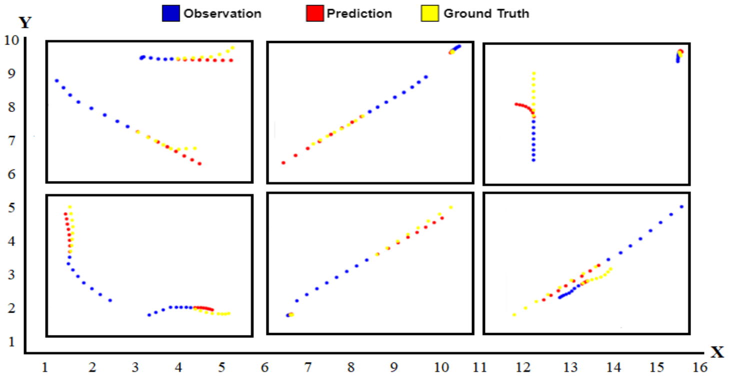

Fig. 15: Future motion trajectories predicted by proposed model

FDE and ADE of proposed methodology with five latest methodologies is shown in figure 16.

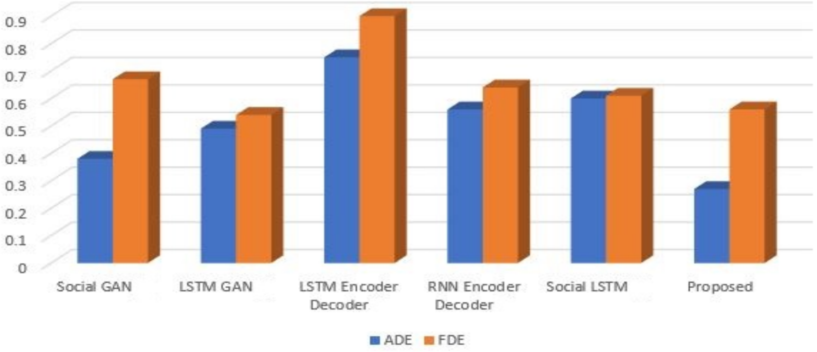

Fig. 16: Comparative analysis with [44] [11] [38] [45] [35]

The comparative analysis shows that proposed methodology outperforms previous methodologies in terms of FDE and ADE measures. The reason why our proposed methodology surpasses state of the art techniques is because of incorporating human-object interaction and human-human interaction along with previous motion trajectories of humans in order to predict the future motion trajectories.

The previous methodologies which are not considering the environmental context have higher rate of collision with objects present in the environment. Table 5 shows the collision of a human with objects in future motion trajectories in 60 minutes video.

The proposed system has least number of collisions.

Methodology	No of Collisions
Agrim gupta, et. al [44]	29
Amir Sadeghian, et al. [11]	21
T. Fernando, el. al [38]	35
Lee, et. al [45]	25
A. Alhai, et. al [35]	16
Proposed	3

Table 5: number of collision of a human with objects in 60 minutes video

Conclusion and Future Scope

The proposed research work provides a fully autonomous system to predict the future motion trajectories of pedestrians in social environment through video stream. It also outperforms the previous techniques in term of final displacement error (FDE) and the average displacement error (ADE). The proposed system achieves 70% ADE and 41% FDE lower error rate than previous methodologies. It further decreases the number of collisions in future motion paths. The reason behind this is incorporation of the environmental context like human-human interaction and human-object interaction with past motion trajectories to predict the future motion trajectories. The future motion trajectories of human is also dependent on the geographical information like the road structure and the position of pedestrians. The future work will focus on incorporation of geographical information with existing features to achieve the better accuracy. It will also focus on the techniques of how to merge the geographical information with existing features. More robust dataset in this domain will provide better learning of the network.

ACKNOWLEDGMENT

Financial support for this study was provided by a grant from the National Center For Artificial Intelligence at University of Engineering and Technology, Lahore, Pakistan. The authors wish to thank Al-Khawarizimi Institute of Computer Science, UET Lahore for providing research platform and technical support.

References

1. M. A. Azad, M. Alazab, F. Riaz, J. Arshad, and T. Abullah, “Socioscope: I know who you are, a robo, human caller or service number,” *Future Generation Computer Systems*, vol. 105, pp. 297–307, 2020.
2. M. A. Azad and R. Morla, “Caller-rep: Detecting unwanted calls with caller social strength,” *Computers & Security*, vol. 39, pp. 219–236, 2013.
3. J. Gambrell and A. P. Aya Batrawy, “New tally shows at least 1,621 killed in saudi hajj tragedy,” *https://www.businessinsider.com/ap-new-tally-shows-at-least-1621-killed-in-saudi-hajj-tragedy-2015-10*, vol. 0, no. 0, p. 0, 2015.
4. P. C. M. Bush, “Police with the latest information on the mosque shootings,” *https://www.rnz.co.nz/news/national/384896/police-with-the-latest-information-on-the-mosque-shootings*, vol. 0, no. 0, p. 0, 2019.
5. N. Master, “Intentional homicide, number and rate per 100,000 population,” *https://www.nationmaster.com/country-info/stats/Crime/Violent-crime/Murder-rate*, vol. 0, no. 0, p. 0, 2010.

6. A. Peltier, Elian; Breeden, "France declares strasbourg shooting an act of terrorism," *https://www.nytimes.com/2018/12/12/world/europe/france-strasbourg-shooting.html*, vol. 0, no. 0, p. 0, 2010.
7. S. Sultan, A. Javed, A. Irtaza, H. Dawood, H. Dawood, and A. K. Bashir, "A hybrid egocentric video summarization method to improve the healthcare for alzheimer patients," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 10, pp. 4197–4206, 2019.
8. C. S. Hussain, M.-S. Park, A. K. Bashir, S. C. Shah, and J. Lee, "A collaborative scheme for boundary detection and tracking of continuous objects in wsns," *Intelligent Automation & Soft Computing*, vol. 19, no. 3, pp. 439–456, 2013.
9. S. Saleem, A. Dilawari, U. G. Khan, R. Iqbal, S. Wan, and T. Umer, "Stateful human-centered visual captioning system to aid video surveillance," *Computers & Electrical Engineering*, vol. 78, pp. 108–119, 2019.
10. A. Ali, H. Rafique, T. Arshad, M. A. Alqarni, S. H. Chauhdary, and A. K. Bashir, "A fractal-based authentication technique using sierpinski triangles in smart devices," *Sensors*, vol. 19, no. 3, p. 678, 2019.
11. A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1349–1358, 2019.
12. B. Kim and J. Pineau, "Socially adaptive path planning in human environments using inverse reinforcement learning," *International Journal of Social Robotics*, vol. 8, no. 1, pp. 51–66, 2016.
13. D. Vasquez, F. Large, T. Fraichard, and C. Laugier, "High-speed autonomous navigation with motion prediction for unknown moving obstacles," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 1, pp. 82–87, IEEE, 2004.
14. M. Z. Khan, S. Harous, S. U. Hassan, M. U. G. Khan, R. Iqbal, and S. Mumtaz, "Deep unified model for face recognition based on convolution neural network and edge computing," *IEEE Access*, vol. 7, pp. 72622–72633, 2019.
15. J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent nn: First results," *arXiv preprint arXiv:1412.1602*, 2014.
16. M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *2010 IEEE International Conference on Robotics and Automation*, pp. 464–469, IEEE, 2010.
17. L. Ballan, F. Castaldo, A. Alahi, F. Palmieri, and S. Savarese, "Knowledge transfer for scene-specific motion prediction," in *European Conference on Computer Vision*, pp. 697–713, Springer, 2016.
18. P. Coscia, F. Castaldo, F. A. Palmieri, L. Ballan, A. Alahi, and S. Savarese, "Point-based path prediction from polar histograms," in *2016 19th International Conference on Information Fusion (FUSION)*, pp. 1961–1967, IEEE, 2016.
19. Z. He and L. Jin, "Activity recognition from acceleration data based on discrete consine transform and svm," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 5041–5044, IEEE, 2009.
20. K. M. Chathuramali and R. Rodrigo, "Faster human activity recognition with svm," in *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*, pp. 197–203, IEEE, 2012.
21. M. H. Bhatti, J. Khan, M. U. G. Khan, R. Iqbal, M. Aloqaily, Y. Jararweh, and B. Gupta, "Soft computing-based eeg classification by optimal feature selection and neural networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 10, pp. 5747–5754, 2019.
22. S. Jiang, M. Lian, C. Lu, S. Ruan, Z. Wang, and B. Chen, "Svm-ds fusion based soft fault detection and diagnosis in solar water heaters," *Energy Exploration & Exploitation*, vol. 37, no. 3, pp. 1125–1146, 2019.
23. O. Gashteroodkhani, M. Majidi, M. Etezadi-Amoli, A. Nematollahi, and B. Vahidi, "A hybrid svm-tt transform-based method for fault location in hybrid transmission lines with underground cables," *Electric Power Systems Research*, vol. 170, pp. 205–214, 2019.
24. S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, "How to approach humans?: strategies for social robots to initiate interaction," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 109–116, ACM, 2009.

25. J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, pp. 2980–2988, 2015.
26. A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, pp. 1764–1772, 2014.
27. D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
28. M. Z. Khan, S. Jabeen, S. ul Hassan, M. Hassan, and M. U. G. Khan, "Video summarization using cnn and bidirectional lstm by utilizing scene boundary detection," in *2019 International Conference on Applied and Engineering Mathematics (ICAEM)*, pp. 197–202, IEEE, 2019.
29. G. Khan, S. Jabeen, M. Z. Khan, M. U. G. Khan, and R. Iqbal, "Blockchain-enabled deep semantic video-to-video summarization for iot devices," *Computers & Electrical Engineering*, vol. 81, p. 106524, 2020.
30. A. Karpathy, A. Joulin, and L. F. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," in *Advances in neural information processing systems*, pp. 1889–1897, 2014.
31. O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.
32. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, 2015.
33. T. Shu, S. Todorovic, and S.-C. Zhu, "Cern: confidence-energy recurrent network for group activity recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5523–5531, 2017.
34. J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *European Conference on Computer Vision*, pp. 816–833, Springer, 2016.
35. A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.
36. Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118, 2015.
37. N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International conference on machine learning*, pp. 843–852, 2015.
38. T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection," *Neural networks*, vol. 108, pp. 466–478, 2018.
39. K. Goel and A. Robicquet, "Learning causalities behind human trajectories," *Conference on Computer Vision and Pattern Recognition*, 2015.
40. H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 169–175, IEEE, 2018.
41. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
42. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.
43. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
44. A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2255–2264, 2018.
45. N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 336–345, 2017.