A NEW FORM OF INTERLOCKING DEVELOPING TECHNOLOGY FOR LEVEL CROSSINGS AND DEPOTS WITH INTERNATIONAL APPLICATIONS

Muhammad A B Fayyaz
PhD 2021

A NEW FORM OF INTERLOCKING DEVELOPING TECHNOLOGY FOR LEVEL CROSSINGS AND DEPOTS WITH INTERNATIONAL APPLICATIONS

Muhammad A B Fayyaz

A thesis submitted in partial fulfilment of the requirements of

Manchester Metropolitan University

for the Degree of Doctor of Philosophy

Science and Engineering Department

MANCHESTER METROPOLITAN UNIVERSITY

In collaboration with OSL Limited

2021

ABSTRACT

There are multiple large rail infrastructure projects planned or currently being undertaken within the United Kingdom. Many of these projects aim to reduce the continual issue of limited or overcapacity service. These projects involve an expansion of Rail lines, introducing faster lines, improved stations in towns and cities and better communication networks. Some major projects like Control Period 6 (CP6) are being managed by Network Rail; where projects are initiated throughout Great Britain. Many projects are managed outside Great Britain e.g., Trans-European Transport Network Program, which is planning for expansion of Rail lines (almost double) for High-Speed Rails (category I and II). These projects will increase the number of junctions and Level Crossings. A Level Crossing is where a Rail Line is crossed by a road or a walkway without the use of a tunnel or bridge. The misuse from the road users account for nearly 90% of the fatalities and near misses at Level Crossings. During 2016/2017, the Rail Network recorded 6 fatalities, about 400 near-misses and more than 77 incidents of shock and trauma. Accidents at Level Crossings represent 8% of the total accidents from the whole Rail Network. Office of Rail and Road (ORR) suggested that among these accidents at Level Crossings 90% of them are pedestrians. Such high numbers of accidents, fatalities and high risk have alarmed authorities. These authorities found it necessary to invest time and utilise given resources to improve the safety system at a Level Crossing using the safer and reliable interlocking system. The interlocking system is a feature of a control system that makes the state of two functions mutually independent. The primary function of Interlocking is to ensure that trains are safe from collision and derailment. Considering the risk associated with the Level Crossing system, the new proposed interlocking system should utilise the sensing system available at a Level Crossing to significantly reduce implementation cost and comply with the given standards and Risk Assessments. The new proposed interlocking system is designed to meet the "Safety Integrity Level- SIL" and possibly use the "2oo2" approach for its application at a Level Crossing, where the operational cycle is automated or train driver is alarmed for risk situations. Importantly, the new proposed system should detect and classify small objects and provide a reasonable solution to the current risk associated with Level Crossing, which was impossible with the traditional sensing systems.

The present work discusses the sensors and algorithms used and has the potential to detect and classify objects within a Level Crossing area. The review of existing solutions e.g Inductive Loops and other major sensors allows the reader to understand why RADAR and Video Cameras are preferable choices of a sensing system for a Level Crossing. Video data provides sufficient information for the proposed algorithm to detect and classify objects at Level Crossings without the need of a manual "operator". The RADAR sensing system can provide information using micro-Doppler signatures, which are generated from small regular movements of an obstacle. The two sensors will make the system a two-layer resilient system. The processed information from these two sensing systems is used as the "2oo2" logic system for Interlocking for automating the operational cycle or alarm the train drive using effective communication e.g., GSM-R. These two sensors provide sufficient information for the proposed algorithm, which will allow the system to automatically make an "intelligent decision" and proceed with a safe Level Crossing operational cycle.

Many existing traditional algorithms depend on pixels values, which are compared with background pixels. This approach cannot detect complex textures, adapt to a dynamic background or avoid detection of unnecessary harmless objects. To avoid these problems, the proposed work utilises "Deep Learning" technology integrated with the proposed Vision and RADAR system. The Deep Learning technology can learn representations from labelled pixels; hence it does not depend on background pixels. The Deep

Learning technology can classify, detect and localise objects at a Level Crossing area. It can classify and differentiate between a child and a small inanimate object, which was impossible with traditional algorithms. The system can detect an object regardless of its position, orientation and scale without any additional training because it learns representation from the data and does not rely on background pixels. The proposed system e.g., Deep Learning technology is integrated with the existing Vision System and RADAR installed at a Level Crossing, hence implementation cost is significantly reduced as well.

The proposed work address two main aspects of training a model using Deep Learning technology; training from scratch and training using Transfer Learning techniques. Results are demonstrated for Image Classification, Object Detection and micro-Doppler signals from RADAR. An architecture of Convolutional Neural Network from scratch is trained consisting of Input Layer, Convolution, Pooling and Dropout Layer. The model achieves an accuracy of about 66.78%. Different notable models are trained using Transfer Learning techniques and their results are mentioned along with the MobileNet model, which achieves the highest accuracy of 91.9%. The difference between Image Classification and Object Detection is discussed and results for Object Detection are mentioned as well, where the Loss metrics are used to evaluate the performance of the Object Detector. MobileNet achieves the smallest loss metric of about 0.092. These results clearly show the effectiveness and preferability of these models for their applicability at Level Crossings. Another Convolutional Neural Network is trained using micro-Doppler signatures from the Radar system. The model trained using the micro-Doppler signature achieved an accuracy of 92%.

The present work also addresses the Risk Assessment associated with the installation and maintenance of the system using Deep Learning technology. RAMS (Reliability, Availability, Maintainability and Safety) management system is used to address the General and Specific Risks associated with the sensing system integrated with the Deep Learning technology. Finally, the work is concluded with the preferred choice, its application, results and associated Risk Assessment. Deep Learning is an evolving field with new improvements being introduced constantly. Any new challenges and problems should be monitored regularly. Some future work is discussed as well. To further improve the model's accuracy, the dataset from the same distribution should be gathered with the cooperation of relevant Railway authorities. Also, the RADAR dataset could be generated rather than simulated to further include diversity and avoid any biases in the dataset during the training process. Also, the proposed system can be implemented and used in different applications within the Rail Industry e.g., passenger census and classification of passengers at the platform as discussed in the work.

ACKNOWLEDGEMENTS

In the Name of God, the One, to whom we shall return and to whom all Praise truly belong.

For the present work, I have to acknowledge my supervisor and mentor Dr Christopher Johnson whose constant advice, guidelines and supervision helped me with my thesis and PhD as a whole. It would have not been possible without him. I would specially like to acknowledge my Supervisor Dr Hong Li, whose constant support, guidance and mentorship helped me bring PhD into its final stage. She was a great help throughout the process of completion, viva and examination. Also, I would like to mention my parents; Mr Fayyaz Hussain and Mrs Kousar Parveen who are light in my life and a source of blessing.

I would also like to thank OSL Limited and Manchester Metropolitan University for providing me with an opportunity to work on this project. I would like to thank them for their constant support and trust.

CONTENTS

FIGURES

TABLES

1 INTRODUCTION

1.1 OVERVIEW

"Interlocking"(How, 2020) is defined as a feature of a control system that makes the state of two functions mutually independent. The primary function of Interlocking is to ensure that trains are safe from collision and derailment. The train industry in its early days did not have any safety mechanism for interlockings but expansion in rail lines demands a safer and reliable interlocking system. Multiple projects are being planned and executed in different regions of Great Britain. These projects directly address the issue of over-population and ever-increasing demand for train services, which our current system is unable to cope with. Some of these major project plans for Control Period 6 (CP6) are mentioned by Network Rail (Network Rail, 2020d). Some key projects include the Cross rail project (Network Rail, 2020a), which will allow 1.5 million more people to travel in Central London within 45 minutes. Another project is the Great North Rail Project (Network Rail, 2020b), which is part of Britain's Railway Upgrade Plan connecting different towns and cities. Forecasts in the Trans-European Transport Network program predict that Trans-European High Speed (HS) network (category I and II Lines) will more than double to reach a length of 22,140 km long by 2020, compared to its length of 9,693 km in 2008. By 2030, this network is expected to comprise 30,750 km of track and traffic to have risen to approximately 535 billion passengers per annum (European Union Agency for Railways, 2017). Such expansions within Rail Industries demands the electrification of the existing systems, and upgrading of these technologies become inevitable. The expansions ultimately increase the number of junctions, intersections and Level Crossings. A Level Crossing is where a rail line is crossed by a right of way without the use of a tunnel or bridge. There are currently 7500-8000 operational Level Crossings within Great Britain (Steven, 2014) and about 7000 are actively used on Network Rail managed infrastructure. Out of these 7000 Level Crossings, around 1500 are present on public vehicular roads and remaining on a public footpath or private roads. With such numbers, the risk associated with a Level Crossing is significantly high, where heavy machinery crosses with high speeds, and road users such as vehicles and pedestrians misuse the Level Crossing during its operational-cycle.

Many Level Crossings do not provide enough information to the users e.g., lack of audible or visual warnings, which lead to misuse of these Level Crossings. The misuse from the road users accounts for nearly 90% of the risk encountered in the previous five years. Level Crossings account for nearly half of the potentially higher risk incidents on British railways. To avoid misuse different precautionary measures should be initiated e.g., providing sufficient time, audible or visual warnings and chemical sprays if users trespass at Level Crossings. For Great Britain, there have been 3 fatalities and 385 near misses at Level Crossings in 2015-2016. Furthermore, the Rail Safety and Standards Board (RSSB) in its annual safety report highlighted the risk of incidents at Level Crossings during 2016/17 with a further 6 fatalities at Level Crossings, including 4 pedestrians and 2 road vehicles. Level Crossings account for 8% of the industry's risk compared to other accidental risk factors within the UK rail's network. In addition to these 6 major incidents, the RSSB recorded 77 minor injuries and 39 shock and trauma incidents occurred due to near misses and 6 collisions between train and road vehicle. RSSB suggested the major factors for most of these incidents are related to the design of the Level Crossings (Office of Rail and Road, 2017). Also, the ORR suggested that approximately 73% of Level Crossings fatalities involve pedestrians (Office of Rail and Road, 2016).

All technology employed in the UK network must satisfy the “Network Rail Assurance Panel Process (NRAP)”, which governs rail processes. When a change is introduced that could affect the risk profile of the Network Rail Infrastructure, NRAP ensures the compliance of the respective engineering processes and technologies with the Network Rail’s responsibility, health and safety systems. The Level Crossing Strategy and Risk Assessment are considered in detail within NR/L1/XNG/100/02 and NR/L2/OPS/100 standards. Furthermore, technologies must be approved through stringent business regulations and standards NR/L2/RSE/100/06 that have reached TRL8 or above. Risk components such as interlocking devices will be reviewed against the NR/L2/RSE/100/06 with approval weighed against the risk to the business. All aspects of any proposed system will be reviewed for redundancy ensuring that even small changes to the software system ensure system fail-safe expectations (Network Rail, 2018a).

To ensure a safer and reliable system at a Level Crossing, the use of Deep Learning is introduced. Deep Learning, which is a subset of Machine Learning is composed of multiple processing layers. These multiple layers learn representations from the large dataset using the backpropagation algorithm, which indicates how a machine should change its internal parameters. These internal hyperparameters are used to compute the representation of a layer from the representation of the previous layer. With enough representation learned, very complex functions are learned and used for applications such as classification and detection. For classification tasks, the higher layers of representations increase the aspects which are required for discriminating the objects and suppress the irrelevant variations(LeCun, Bengio and Hinton, 2015). The ability to learn itself and classify the objects without the intervention of a manual operator makes the use of Deep Learning the most appropriate choice for a Level Crossing application. The Deep Learning algorithm is integrated with the existing sensing system, which further reduces the cost and justifies the Risk Model for the new application proposed at Level Crossings.

The Deep Learning approach is integrated with both preferred choice of sensors e.g., Vision and RADAR system for an effective interlocking system. The “two out of two” approach has two processing channels e.g., Vision and RADAR and each channel are processed and output executed is according to the given industry standard. These two outputs are compared and if they do not match, the system is shut down to avoid failure mode. Such an approach will ensure “Safety Integrity Level -SIL” of 3 or 4, where the underlying principle is “fail-safe”. Fail-safe does not mean the system is 100% safe rather the ability to signal and stop the train in risk situations.

1.2 MOTIVATION

Current Level Crossings systems are unable to deal with the high demand of road and rail users also requiring the manual intervention of a signaller to notify the train driver for any high-risk situations during the operational cycle of Level Crossings. Such a system demands an upgrade in the technology to automate the process and provide reliable results. This upgrade should justify the cost with the risk reduction and aim to eliminate the use of “manual monitoring” except in degraded modes of operation. Many different works have been proposed to utilise different sensing systems as discussed later in the thesis along with their associated algorithms. Most of these proposed systems are traditional and have their limitations for their applicability at a Level Crossing site. The proposed work will use Deep Learning technologies integrated with the existing sensing subsystems to automate the process of classifying and detecting objects during a Level Crossing operational cycle. The detection of objects in real-time will allow the system to automate the Level Crossing inspection process and start an operational cycle or else alarm the train driver using an effective communication channel e.g., GSM-R in case of a high-risk situation. The discussed Deep Learning technology does not require a high cost of installation or removal of existing sensing subsystems at Level Crossings. In purpose, it provides a safer and reliable system compared with the traditional methods and algorithms used.

Many heavily used Level Crossings have automated operational cycles, where the cycle is operated using fixed periods. The cycle starts with certain audible or visual warnings and after a certain timeframe, the barriers are lowered down to let the train pass and later barriers are returned to their original position for level Crossing users. However, the mentioned operational cycle does not respond to high-risk situations and rely on a manual operator or signaller to alert the train driver. The motivation for the present work is, therefore, to automate the operational cycle at Level Crossings and respond to high-risk situations without the need for a manual operator and signaller. The automation process should reduce the risk associated with its users and give reliable results for risk analysis. The new system should not significantly increase the cost of installation and maintenance. The motivation for using Deep Learning technologies are as follows:

1. *Application of new novel ideas to Railway Industry*

The traditional approaches for a Level Crossing or elsewhere within the Railway Industry mostly rely on the manual operator or a signaller to respond to high-risk situations, risking the safety of Level Crossing users and passengers on the train. The traditional approach relies on algorithms where an obstacle is detected from the difference in background and new pixels or certain specific properties e.g., speed and direction obtained from the Radar system, which is not suitable for the highly dynamic and complex environment of a Level Crossing. Therefore, the integration of Deep Learning technology with the existing vision system is introduced which can learn representation from the data regardless of its orientation, size and position within the Level Crossings area. The Deep Learning model will classify and detect obstacles in real-time to analyse and respond to high-risk situations. Deep Learning technology is highly adaptable and has the potential to be applied in other areas of the rail industry. Cascading of solutions from busy Level Crossings with CCTV to User work Crossing (UWC) provides a feasible rollout where techniques can be adapted to support platform safety and trespassing issues. As Level Crossings are fixed assets and provide some level of a controlled environment it would seem an appropriate starting point for their application.

2. *Reducing Railway workloads*

With the increasing number of Level Crossings and population, the associated risk is ever-increasing, which means an increase in manual “operators” and human intervention if relied on traditional approaches. However, the reaction time and statistics from past years provide a clear motivation for a more reliable and safer system. The training, protective measures and the recruitment of manual operators increase the cost and time for the Railway Industries. All these drawbacks are avoided with this novel application of Deep Learning technologies at Level Crossings, which eliminate the need for manual operator and provides more reliable quantifiable results for detection and risk analysis. Deep Learning technologies will allow the Rail industry to utilise this “workforce” elsewhere in Industry.

3. *Utilising new Data*

Computer Vision techniques used to classify and detect objects is ever increasing because of the availability of the data. This open-source dataset is used to train the model required for the prediction and classification of objects for Level Crossing applications. Once trained and deployed for its application, the railway users can further increase the accuracy of the network by labelling data particular for a Level Crossing (data from the same distribution). The ability of Deep Learning to learn and re-learn from the available data is another motivation of this work and its applicability at Level Crossing sites.

For the Radar system, a new dataset using micro-Doppler signals are simulated to train a Convolutional Neural Network. The dataset will be made public so more research and improvement to the given model is achieved. The ability to learn representation from a simulated dataset using micro-Doppler signatures and achieve accurate results is another motivation of using Deep Learning for its applicability at Level Crossings.

1.3 Research Questions and Objective

The research question addressed in this thesis is proposed by OSL Rail & Engineering Services and Manchester Metropolitan University. This research aims to provide an answer to the following:

"Do the current state of the art safety devices provide adequate coverage for High Speed as part of the UK development strategy?"

&

"Can modern developments in engineering and technology produce additional novel interlocking methods for Level Crossings and depots?"

The research aim is further elaborated in the given objectives, which are more explanatory.

1. *Determine the most appropriate sensing system for a Level Crossing Application*

Different sensing systems are utilised and proposed over recent years. The proposed work will provide a survey on these available sensors and select the most appropriate sensing system, which ensures safety for High Speed and Depots Environments.

2. *Development of a two-layer sensing system*

The development of two sensing systems, which is capable of detecting any failures and provide a fail-safe mechanism for any new technology enhancements without interfering with the normal operation of a Level Crossing is proposed. The "2oo2" interlocking system provides a fail-safe mechanism, where train drivers are alarmed in high-risk situations.

3. *Determine the most appropriate algorithm for detection and classification of obstacles at Level Crossings*

Different algorithms are used and proposed to detect obstacles at Level Crossings during its operational cycle. The ability to detect obstacles from traditional algorithms will only facilitate the manual operator to make better judgments. However, the proposed algorithms can effectively classify and detect obstacles with no intervention of manual operators and provide more accurate and reliable results compared to traditional approaches. The fail-safe mechanism can automate the operational cycle at a Level Crossing without the need of a manual "operator".

4. *Ensure that the applied system is applicable both in theory and practise*

The proposed work is trained using a labelled dataset and its predictions on actual images from a video stream at the real site within Great Britain from a Level Crossing are mentioned. The algorithm can process and analyse the data using low computation power available at Level Crossing's site to detect and classify obstacles for automation. Also, the system can recover from any failure point during its operation.

5. *Ensure the applied sensing system and algorithm outperforms the traditional approaches.*

The traditional sensing systems and their associated algorithms have their limitations. The new proposed system will significantly reduce these limitations and make the system safer and reliable. The results from Deep Learning techniques are compared with traditional algorithms, for example, the Deep Learning model does not depend on background pixels values and can discriminate between a harmless object and a child present at a Level Crossing.

1.4 Research Challenges

To achieve the proposed aim and objectives several challenges were raised during the project and these issues were addressed and appropriate measures were suggested.

1. *Scepticism Regarding Automated Techniques*

The use of Deep Learning technology as proposed in this work can fully automate the process of a Level Crossing operational cycle. However, the Rail Industry is very sceptical of using a “machine” and trust it, compared with their traditional method of relying on a manual “human” operator. Deep Learning technology shows better results compared with the human error rate. The proposed work will show visual results and statistics to convince the Rail industry for an upgrade and use of “machine” compared with the traditional use of manual “human” operator. The output results are visualised using different software’s e.g., tableau, which can help relevant authorities to analyse data for Risk Assessment as well.

2. *High-Level Performance through an Appropriate Deep Learning technology*

There is a range of methods that are used in Deep Learning technology to classify and detect an obstacle. These methods differ in their model, techniques and data they use for training. To achieve high-level performance from Deep Learning, the practitioner should be able to choose the right model with the right data and training to achieve more reliable results. The proposed work utilises most models; Models from Scratch and Models using Transfer Learning techniques to demonstrate and visualise the results. To justify the implementation of Deep Learning, the results achieved using Deep Learning technology must be better and more reliable than any previously proposed manual method of detection. The other traditional algorithms have their limitations and are unable to detect certain situations e.g., detection of small objects or false alarms from a dynamic environment. The proposed Deep Learning techniques should outperform these traditional approaches, especially in these given scenarios.

3. *The complexity of the RADAR system*

Two sensing systems are used for the proposed work; CCTV and RADAR. The operation of the video sensing system is not complex and the dataset available from CCTV is in abundance because of the rise in the Deep Learning field. However, RADAR is a complex system with many different applications and requires different post-processing techniques. The integration of Deep Learning in RADAR is still in its early stages, hence the data available to train a Deep Learning model is complex compared with the CCTV sensing system. The micro-doppler signals from RADAR are used for this particular work for training a Deep Learning model. Some classes need defining to simulate their data and model the class. The simulated dataset is used to train the model and the dataset is made public for further research and contribution to the research community.

4. *Risk Assessment*

New technology or techniques introduced within the Rail Industry must comply with the stringent business and industrials rules and standards. The proposed work attempt to address these standards and risk assessments using the RAMS managed system. The general and specific Risk Assessments are mentioned and discussed in the work to facilitate the relevant authorities. The proposed work addresses all these challenges and any other challenges along the work to give appropriate solutions.

1.5 Research Contribution

From the discussed objectives and challenges, the thesis presents a contribution to the Rail Industry. The research contribution is as follows:

1. Traditionally, the proposed algorithm relied on pixel values to model a background, which was compared with new pixel values to detect the presence of foreground. The approach was limited to the static background with the least change in light and conditions. To overcome such limitations, an effective Deep Learning algorithm is proposed, which automatically learns the representations from the dataset to classify and detect objects and does not require a manual feature extractor used in the traditional approach as discussed in Deep Learning.
2. Traditional algorithm and post-processing techniques could make a rough estimation of an object by using common properties e.g., size and shape. Such an approach had certain limitations to detect small objects and classify between harmless with harmful objects at a Level Crossing. The proposed Deep Learning can effectively detect and classify these objects, which makes the system safer and reliable as discussed in *Traditional Algorithms*.
3. The traditional provide limited information, hence a manual operator or a signaler is required to analyze the information and make decisions to either start the operational cycle or alarm the train driver if necessary. Hence, the system relied on manual "human intervention" for high-risk situations. This makes the system slow and unreliable since it relies on human response time. The proposed Deep Learning can automate the operational cycle and does not require the intervention of the human operator.
4. The relevant authorities within the Railway Industry collect data for their Risk Model to take precautionary measures and suggest relevant improvements for a Level Crossing application. The data collection is a separate task and is not achieved from the traditional approaches used at Level Crossing systems. The proposed Deep Learning technique can store the relevant information during its operation and update the system at the required time interval. The data provides a detailed statistic of users at Level Crossings, which is vital information for the Risk Model and Assessment.
5. The dataset obtained using the micro-doppler signal from RADAR is defined and simulated in MATLAB using high-level functions. No such data is available online or for research, the obtained dataset in this research is used to train the Neural Network using Deep Learning techniques and will be shared with the public. This dataset will act as a foundation for future research and its applicability at Level Crossings or other platforms using RADAR.
6. A fail-safe mechanism is introduced using a two-layered sensing system, CCTV and RADAR. The Deep Learning technique is integrated with both sensors and works well with low computation power, enough to process 15fps. The 15fps rate suggests that even a certain false-positive at any

moment will be superseded by many positive classifications before a human eye can spot such an error. This makes the system more robust and safer compared with the previous traditional approach. The “2oo2” approach will ensure a safer Interlocking system with “Safety Integrity Level- SIL” of levels 3-4 providing a fail-safe mechanism.

7. The proposed Deep Learning technique can adapt to different applications. The same model can predict and categorise the passengers on the platform, which is another key area of research. With small changes in the proposed network, the model can categories passengers with bicycles or luggage. A similar approach is used to train a model for behavioural prediction, where any high-risk situations are predicted from uncertain patterns at platforms.

1.6 Publications

The research was presented to different conferences and journals, some of which are mentioned below.

- A review of the technological developments for interlocking at level crossing
- Object Detection at Level Crossing Using Deep Learning
- Object Detection at Level Crossing using Deep Learning Techniques

1.7 Research Methodology

The present work focuses on operational cycles at a Level Crossing and introduces new technology and innovative methods to ensure a safer and reliable interlocking system. The proposed method should justify costs and effectively reduce high-risk situations and provide sufficient data for risk analysis. The acquired data can help relevant authorities to take more effective precautionary measures for up-gradation and expansion of the existing system within the Rail Industry.

To select the most appropriate sensing system, a detailed survey was carried out for sensors. These sensors are currently used or have the potential for their applicability at Level Crossings. The limitations and preferability of each sensor are discussed and the best possible combination of sensors e.g., CCTV and RADAR are proposed. The sensors require some post-processing techniques to analyse the data and make predictions to classify objects to facilitate the manual operator to operate or respond to high-risk situations at a Level Crossing. The traditional and manual algorithms mostly rely on the concept of subtracting pixels from background pixels to detect the foreground. The ineffectiveness of this approach and its limitations are discussed and compared with the Deep Learning technology, which is proposed for the application in the present work. Deep Learning technology can learn features automatically to classify and detect an obstacle and does not rely on pixel values. Deep Learning techniques can learn representation regardless of its size, shape, orientation and position at a Level Crossing area, hence it is adaptable to the dynamic and complex environment at a Level Crossing. A detailed discussion on Deep Learning is presented before its implementation for its application at a Level Crossing.

Different Deep Learning techniques and models are used for classification and detection depending on its application and availability of the data. These techniques use the pre-labelled dataset to train the Neural Network-NN for image classification. Two different methods are available to design and train the Neural Network; Designing and Learning from Scratch or using Transfer Learning techniques. The proposed work will use both of these methods to compare results and analyse which particular method is suitable for its application at a Level Crossing.

The adapted techniques and approaches for Deep Learning are as follows:

1. A Neural Network is designed and trained from scratch using a custom dataset downloaded from open-source (ImageNet) for the Classification of objects at a Level Crossing.
2. Image Classification results obtained from a model trained from scratch is compared with the Neural Network trained using Transfer Learning techniques using pre-trained models.
3. Object Detectors are trained using transfer learning techniques which provides results e.g., Classification, Localisation and Detection for multiple objects at a Level Crossing.
4. A dataset obtained using Doppler Signatures from RADAR is fed to Neural Network to add another layer of resilience to the system.

The trained model is evaluated using a “test-dataset”, which is unseen data for the model. The data correspond to the real-site of Level Crossings within Great Britain. Results from these two channels ensure a “2oo2” interlocking system, which ensures high “Safety Integrity Level-SIL”. High SIL of 3-4 means the system is fail-safe and respond effectively to high-risk situations. Risk Assessment for the given sensing system for its installation and maintenance is discussed. Finally, the work suggests some key areas where the given technology can be integrated within the Rail Industry. Some common applications include categorization of passengers at the platform, analyzing the pattern and behaviour of passengers or census of users at a particular platform or train service.

1.8 Structure of the Thesis

The brief introduction helps the reader to understand the problem addressed in this work. The present work is divided into different segments for easy reading and referencing. The work starts with a detailed introduction in “Literature Review”, where “Introduction” talks about the problem associated with Level Crossings and the need for an alternative solution to upgrade the existing technology for detection and automation of the operational cycle. Before finalizing the sensor and its associated algorithm, the present work discusses current sensors and algorithms deployed at a Level Crossing in “Obstacle Detection System” and “Obstacle Detection Algorithms” respectively. The history of the Interlocking System and its relevance with the proposed work in Level Crossings are discussed in “System”. Once the CTTV and RADAR are finalized, they are discussed in “CCTV” and “RADAR” respectively, also, the proposed algorithm Deep Learning is discussed in detail in “Algorithm” along with its history and notable works.

Before the application of this proposed system, the present work discusses the associated Risk Assessment. The work uses CCTV and RADAR to collect the data for training the network as discussed in DATASET- CAMERA & DATASET- RADAR. The dataset is used to train the model for Classification discussed in Image Classification using CCTV, detection as discussed in Object Detection using CCTV similar application using RADAR as mentioned in Application of RADAR. Some other achievable applications using the same model is discussed in Other Applications. Finally, the work proposed a “2oo2” Interlocking system for its application at a Level Crossing in Interlocking System before giving some Conclusive Remarks and Future Work in Conclusion & Future Work. The work ends with some proposed future work in “Future Work” followed by References & Appendix.

2 LITERATURE REVIEW

2.1 INTRODUCTION

Different types of Level Crossings are available at different sites depending on the traffic volume, road users, associated risk and cost justification as outlined by the Office of Rail Road (ORR). There are about 10 different types of Level Crossings mentioned in (Office of Rail and Road, 2011), where the most frequent and of point of interest for researchers (for reasons discussed later) are Level Crossings with automatic barrier and obstacle detection technology. Regardless of the type of Level Crossing, it should have certain visual or audible warning signs for road users and rail operators. These warning signs allow the users to either warn or confirm if the Crossing area is clear for the train to pass through. Some Crossings may have gates or others half or full barriers to block the Crossing during its operational cycle. The operational cycle is either monitored and operated via an automatic cycle when the train passes the “strike-point (a certain distance from the Level Crossing area)” or via railway signaller at a site or a remote location. From a wider perspective, these different types of Level Crossings within Great Britain are categorised into three different types; Automatic, Passive and Railway Controlled. The information given below in Table 1 is a summary of a detailed report from ORR (Office of Rail and Road, 2011).

Type of a Level Crossing	Operational Cycle
Automatic	No railway staff is required. The train crosses a certain “strike-point”, which automatically starts the operational cycle. The warning signs are initiated to warn road users and in almost 27 seconds the barrier is closed, the train passes and the barriers open again for road users.
Passive	No railways staff or warning signs are in operation. The safety is dependent on the road users, who are often given a telephone line to confirm the arrival of the train.
Railway Controlled	A railway staff either on-site or at a remote location is present. The operator manually checks for any obstacle present before signalling the train driver for a clear pass. New developments within this type of Level Crossing are including an Obstacle detector as well, where the detector results are further verified by the operator before signalling the train driver.

Table 1A summary of "Types of Level Crossings" currently installed within Great Britain.

Since Level Crossings are mostly managed and monitored by Network Rail, the layout of these Level Crossings provides a clear idea of their mapping across Great Britain. Figure 1 shows the layout of 7000 Level Crossings across Great Britain owned and managed by Network Rail.

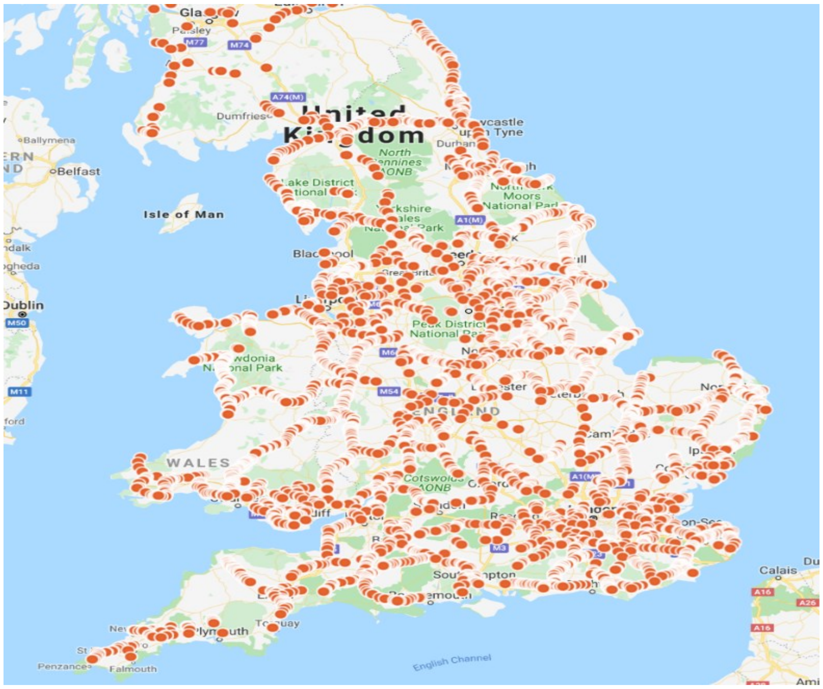


Figure 1Distribution of Level Crossings across Great Britain managed and monitored by Network Rail. Around 7500 Level Crossings are currently present at Great Britain and about 7000 are just managed by Network Rail, hence the map for Network Rail's Level Crossings.

With such high numbers and diversity of these Level Crossings, the risks associated with their use is inevitable resulting from a high train's speed to misuse from road users such as pedestrian and vehicles. Also, several changes are to be expected within the Great Britain train industry such as with increasing population density, increase in train services and platforms. These changes will further increase the risk unless some long-term strategies are not implemented at a Level Crossing. The most effective way to reduce or even eliminate the risk of a Level Crossing is to completely remove one. However, with the given number and diversity of these Level Crossing sites, it is quite impossible at times to remove these Level Crossings especially "Public footpaths Level Crossing". Also, the law states that if a Level Crossing is removed an alternative should be provided such as tunnels or bridges, which is often more expensive and impossible with the difficult and uneven terrain of Great Britain. The construction of bridges or tunnels requires a large capital and time frame as well, which is not justified from the Health and Safety at Work Act etc 1974. The Act requires the relevant organisation such as Railways to reduce risk "as far as it is reasonably practicable". Since the risk associated with a Level Crossing is not justified for its closure, the relevant authorities must look for other alternatives to ensure a safer and reliable operation at a Level Crossing site. Relevant authorities like Network Rail, Office of Rail and Road (ORR) and Rail Safety Standard Board (RSSB) strongly recommend to "investigate, trail and implement new technology, processes and techniques that improve safety through either cost or provision of enhanced protection"(Network Rail, 2019a).

A survey from Network Rail gives numbers of fatalities from the year 2008-2018 in Figure 2. Figure 2 clearly shows the numbers have a non-linear relationship, which demonstrates that no safe and effective mechanism is installed to ensure a safer and reliable operation at a Level Crossing. On a yearly average, there are about 5 public fatalities and 2-3 fatalities for vehicle occupants since the last 10-year time frame.

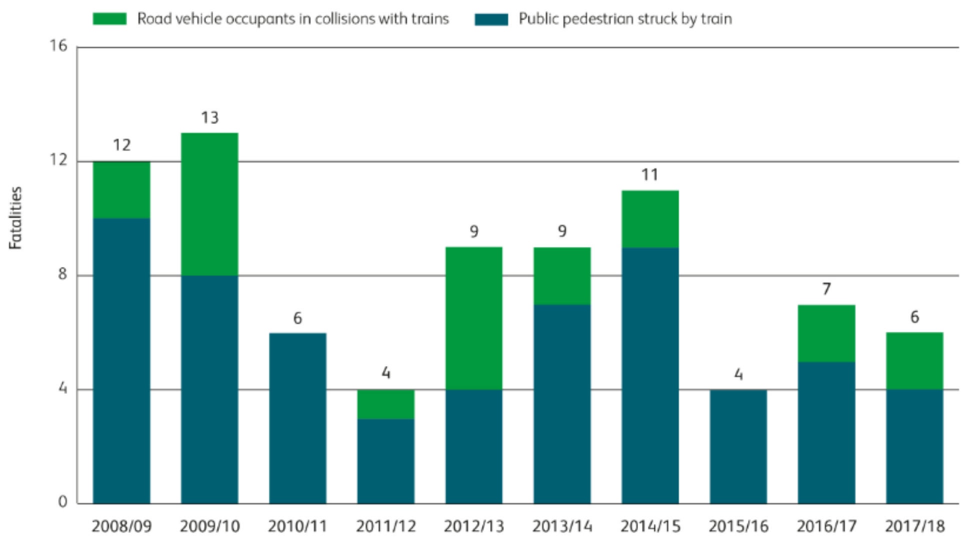


Figure 2Number of fatalities for road vehicle and public pedestrian over the 10-year time period from 2008-2018. Survey courtesy of Network Rail.

In another report by Network Rail, it mentions that the "near-misses" reaches 80 vehicles and about 314 with non-vehicles on average from 2014/2015 till 2018/2019. The number of misuses is high with about 30 reported cases in just one year and 9 cases of collisions with road vehicles in the same one-year time period. Table 2 mentions the numbers as given by Network Rail.

	2014/15	**2015/16**	**2016/17**	**2017/18**	**2018/19**
Level Crossings misuse (MAA) Network-wide	25.46	25.92	29.85	31.38	30.23
Collision with road vehicles	7	4	6	7	9
Train striking pedestrian	12	5	11	7	3
Near miss with road vehicle	76	72	95	80	73
Near miss with non-vehicle users	236	256	276	314	308

Table 2A detailed statistic by Network Rail, which mentions the misuse of the Level Crossing to collisions and near-miss with vehicles and non-vehicles users from 2014-2019.

At the outset, it is useful to compare the performance of Great Britain Level Crossings with European statistics. Basic data on the railway safety performance for each country of Europe, labelled 'Common Safety Indicators' (CSIs) are reported by national authorities annually to the European Union Agency for Railways (ERA), and published by the Agency. The CSI is a major source of data for comparing the performance of Level Crossings in European countries. The data in Table 3 includes the number of Level Crossings and casualties by type (Evans and Hughes, 2019).

Fatalities	**UK**	**Rest of Europe (RoE)**	**UK/RoE**
Level Crossing user fatalities per year	6.4	294.6	
Level Crossing user fatalities per year per million population	0.098	0.580	0.170
Level Crossing user fatalities per year per thousand rail route-km	0.396	1.370	0.289
Level Crossing user fatalities per year per thousand Level Crossing	1.015	2.715	0.374

Table 3The table compared the fatalities of the UK compared with the rest of Europe. It uses an index value as explained in (Evans and Hughes, 2019).

As mentioned earlier, the authorities are concerned with the high risk posed at a Level Crossing from both train and road users. These authorities use the “All Level Crossing Risk Model (ALCRM)” to understand the risk better, so they can focus and plan their investment to further improve the safety and reliability of a Level Crossing as practically as possible. But in reality, the risk models or any laws relating to Level Crossings are as quoted by Department for transport “…too complex… outdated and unclear once found. This can make effective management of a Level Crossing difficult and give rise to safety concern” (Steven, 2014). However, the aim of this research is not to reform the law rather to work on areas that could help improve the safety and risk concerns under the given policies and strategies acceptable from mentioned relevant authorities. And since the mainline network within Railway Industry is owned and managed by Network Rail, it would be necessary to review its policies and strategies related to the Level Crossing Safety.

Network Rails proposed strategy focus on four “E’s”; Education, Enforcement, Enablement and Engineering. Network Rail must continue to increase the awareness for Education, and it could only be carried out effectively if we know what particular category of road user is misusing the Level Crossing. For example, a particular type of a Level Crossing may have a high number of the bicyclist who misuses the Level Crossing while at other Crossing pedestrians are misusing it. Only if relevant authorities have such data then they can effectively promote education about risk mitigation and possible measures towards it. Another important area of research is Engineering; a safer and reliable Level Crossing by introducing new technology justifying the cost along with its safety and performance. Also, Network Rail plans to continue research, review existing technologies and make recommendations in these areas of interest (Network Rail, 2019b).

According to the standard GI/RT7011 by Railway Group Standard, the authority should have a documented risk assessment suited for each type of Level Crossings, in particular, it should cover characteristics of train operations, characteristics of each road user, misuses and effective measures taken to ensure its safety. These among other details would allow authorities to gather data and visualise these statistics to analyse and effectively take measures for Level Crossings safety and reliability. Also, if any changes are brought to any category such as train, frequency of services, type of Level Crossing or road users the Risk Assessments should be reviewed and updated accordingly. The GI/RT7011 section C2.3 suggests "if the risk is sufficiently high for it to be reasonably practicable to do so, the crossing shall be upgraded to a more protective type" (Woolford, 2002). Accordingly, the proposed research will discuss a new technology, which is reasonable and justified for its cost, implementation and effectiveness. It will provide required statistics for risk assessments and upgrade the existing methods of operations for a Level Crossing.

From the given statistics earlier, it can be concluded that the risk posed at a Level Crossing is significantly high and it is reasonable to assume for most cases, it would be impractical to close the Level Crossing. Hence, the focus of research should be around new innovative technologies and operational methods that could justify its cost and effectiveness. The proposed work utilises the existing sensing subsystem and updates the operational method that could reduce the need for a manual signaller or operator at the site or remote location. The same system could gather the statistics and census for categorising those that are misusing the Level Crossings. Since the proposed system can update the operational cycle, eliminate the use of traditional manpower and also provide enough data for risk assessment it would significantly reduce the cost and improve the safety and reliability of the system. To continue discussing the proposed idea, it is essential to discuss the Interlocking system before discussing the traditional and existing sensing systems and their associated algorithms (if any, which is used for post-processing the data) that could potentially be used for obstacle detection at a Level Crossing. Interlocking System will discuss the history of Interlocking and an effective proposed interlocking system before discussing the sensors and their associated post-processing techniques in Measurement Methods.

2.2 Interlocking System

"Interlocking" (How, 2020) is defined as a feature of a control system that makes the state of two functions mutually independent. The primary function of Interlocking is to ensure that trains are safe from collision and derailment. Interlocking does not check that everything is safe for the passage of the train, hence the industrial control system is designed to meet the "Safety Integrity Level (SIL)" to signal when it is safe for train passage. The interlocking functions are associated with the highest SIL of 3-4, where the underlying principle is "fail-safe". Fail-safe does not mean the system is 100% safe rather the ability to signal and stop the train in risk situations.

The train industry in its early days did not have any safety mechanism for interlockings but soon with time accidents happen and the need for a safety system was realised. The first safety mechanism was purely "mechanical" where levers were used for signalling and interlocking, which was replaced with the "electro-mechanical" system. In electro-mechanical systems, the lever would not move unless the lock was energised, which means that other levers were in the right position. All mechanical locking systems were soon replaced with "electrical locking" which did not require any levers or physical force to operate them. The control panel would allow the signaller to check if the track is cleared of other trains and signal the next route if it was safe to do so, all from the switches on the control panel.

Relays were the central part of the "electrical interlocking" system, where different types of relays are used depending on the geographical location and industry. With the emergence of micro-processor and

Programmable Logic Controllers (PLCs), these relays were replaced with the “software-based” interlocking e.g., “Solid State Interlocking” in the UK (Cribbens, 1987). A route relay interlocking is a hard-wired parallel logic machine, where potential failures are well understood and by good design, practice and “fail-safe” mechanisms, the failure modes are very low. Comparatively, computer-based interlocking is very complex and unpredictable and their root causes are very difficult to trace, hence the failure modes are high. The computer-based interlocking is very industry-client specific and required specific design engineers. The safety of the railway is critically dependent on the product and correctness of the specific application, hence so much effort is made into the testing and verification phase. Most of these testing and verification is automated now so engineers can focus more on unusual elements or schemes, which is not automated. To ensure the safety of the system through computer-based Interlocking, a “two out of two (2oo2)” approach is used. The “two out of two” approach has two processing channels and each channel is processed and output is executed according to the given industry standard. These two outputs are compared and if they do not match, the system is shut down to avoid failure mode.

However, the ability of a comparator to simply compare two outputs is not an efficient and reliable approach since both processing units are processing the same task and they both could predict wrong output at the same time. Therefore, the need for a different approach is evident, which

1. Uses different hardware/software for the two channels to reduce the likelihood of common-mode failure.
2. Use more complex cross-checking of states, inputs and outputs of these two channels.
3. Use more than one mechanism to enforce a shutdown mechanism.

If the relay-based interlocking is faulty, it will only affect some train services, whereas, if the computer-based interlocking detects a fault, the system will auto-shutdown and train services for the whole control area are stopped. Therefore, the “two out of three (2oo3)” (Edvard, 2014) approach is suggested, where the output of the majority (two out of three) is executed for the action. However, the issue remains that both of these processing channels are using the same software, they both could predict wrong output at the same time. The most commonly used approach is “duplicated two out of two (2X2oo2)”, where two pairs of processing channels are present and if one finds faults the other pair is at standby for its operation.

Once the sensing sub-system is configured according to the Industrial standards, the need for communication of the information to the relevant platform is required e.g., train driver. Three main interferences are required between Interlockings and other subsystems e.g., Control Desk, Trackside equipment and neighbouring interlockings. The control desk acts as two-way interference, where information is communicated between the control desk and interlocking and vice versa. The information is often automated e.g., Automatic Route Setting (ARS) to relieve signallers with routine work. The trackside equipment is normally connected to nearby object controllers, which provide power and receive two-way information from and to the interlockings. For the interference between neighbouring interlockings, a high integrity data link is required for their communication of information. The idea of physically discrete interlockings is starting to change; the communication between track equipment and train drivers using GSM-R will provide a more effective way of communication for safer and reliable interlocking. Cloud technology could be used for more centralised communication systems or more distributed Interlocking functionality for communication across different interferences within the training Industry.

The present work will use the “2oo2” approach for the Interlocking system, where two processing channels are the Vision System and RADAR system both integrated with the Deep Learning technology. The outputs are processed using different Neural Networks and they both have different hardware configurations, which means they are mutually independent in their operation. The present work also suggests the use of GSM-R to communicate the information from the processing channels to the train driver. This will ensure a safer and reliable interlocking system with a “fail-safe” mechanism. Therefore, it is essential to discuss the sensing sub-system and its associated post-processing techniques before choosing the preferred sensors for their applicability at a Level Crossing. The integration of Deep Learning technology with the preferred sensing system is mentioned as well after discussing the Deep learning technology itself along with its history and technical aspects.

2.3 MEASUREMENT METHODS

It is essential to discuss the state-of-art within sensing systems utilized at a Level Crossing or have the potential for detecting the objects. The obstacle detection system for Level Crossings is categorized into two main types; intrusive or non-intrusive. Intrusive sensors are installed under the tarmac or attached to the rail lines, which naturally makes the installation and maintenance very expensive and hard. The constant wear and tear shorten the product lifecycle and regularly disrupts the train operations. The non-intrusive sensors are installed outside the rail tracks; hence it does not disrupt the rail operations during installation and maintenance. It generally has a longer product-life cycle (Darlington, 2017). The primary choice of sensors for its application at Level Crossings were intrusive sensors such as inductive loops but with years it was entirely replaced with non-intrusive sensors for obvious reasons mentioned earlier. However, it is still necessary to discuss intrusive sensors to understand the demand for non-intrusive sensors.

The Obstacle Detection technologies at a Level Crossing should detect an object (any vehicle or pedestrian) within the Level Crossing area. However, the successful implementation of such a sensing system requires effective communication between the signaller/operator (or sensing system) and the train driver. Effective communication will ensure a safer and reliable interlocking system. The technology must check the area before it starts the process of closing the barriers and signalling the train operator. The new or improved technology must detect objects with more precision and within a shorter time frame. To avoid any misuse of such systems, it should only be implemented at a “Full Barrier” Level Crossing, since the road users will misuse the other types of Level Crossings, as some free way to cross the rail lines is still available. Therefore, the automated Obstacle Detection technology should be implemented at an existing “Full Barrier” Level Crossing or others should be upgraded to “Full Barrier” Crossing before automating the process using this technology. To illustrate the above-mentioned strategy a decision tree is given in (Dent and Marinov, 2017) as mentioned below in Figure 3.

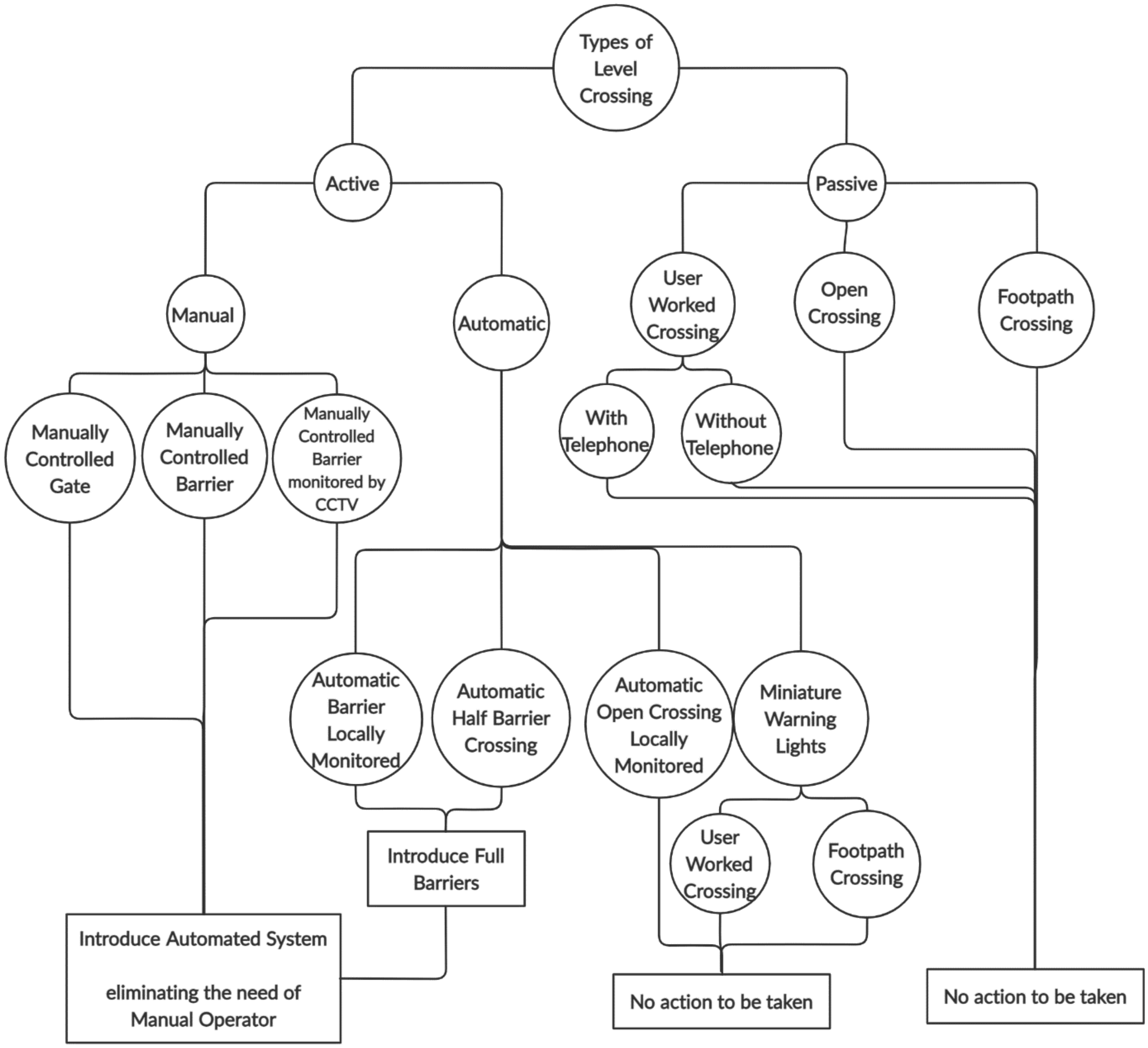


Figure 3A generic flowchart given by (Dent and Marinov, 2017), which shows what should happen at each type of a Level Crossing within the UK.

Many different traditional methods are used for communication between a signaller/operator (or sensing system) and the train driver. One such method was to use Automatic Warning System (AWS), where the AWS system would provide a train driver with audible and visual warnings to caution the driver. Information is conveyed by electromagnetic induction to the moving train through equipment fixed in the middle of the track. The receiver is attached to the end of the train (Railsigns, 2020). AWS was completely replaced with Train Protection & Warning System (TPWS). The TPWS system does not prevent the Signal Passed at Danger (SPAD), rather mitigate the risk by automatically initiating a brake demand and preventing the train from reaching a conflict point ahead of the signal (Apheby, 2016). For the applicability of TPWS, every single Level Crossing would require one TPWS, hence it is not a cost-effective solution.

To ensure safer, reliable and efficient operation, National Rail Telecom (NRT) provides the whole rail industry within Great Britain the telecommunication capability. It delivers the telecommunication capability using Network Rail's telecommunication network, systems and assets e.g., Level Crossing telephones or CCTV. They provide telecommunication capability using IP telecoms Networks (FTNs) or Global System for Mobile Communications- Railway (GSM-R) (Network Rail, 2018b). The automation of communications between track and train can reduce the risk on the railway system. Effective communication will ensure the train driver is alarmed or continuously updated for the presence of obstacles within a Level Crossing area during its operational cycle. For example, the European Train Control System (ETCS) is a train control, signalling and train protection system currently being deployed within Europe (Endersby, 2016). ETCS has three main implementation Levels, where Level 2 and Level 3 provides a continuous communication system using GSM-R radio in the train cab. This provides effective communication between track and train ensuring a safer and reliable operating system. All Levels of ETCS offer enhanced protection when compared with standard UK protection system (AWS and TPWS), hence the Department of Transport is introducing plans to fit 72% of the UK's rail infrastructure with the ETCS by 2038.

The present work integrates the Deep Learning technology with the sensing system to detect and classify obstacles at a Level Crossing area. Once detected, the information should be effectively communicated to a train driver through the ETCS system using GSM-R radio. The information will ensure that the operational cycle at a Level Crossing is carried out effectively and if required, the train driver is alarmed for the high-risk situation to provide him/her with enough time to stop.

The proposed idea of the interlocking system starts with a sensor, therefore, to understand what particular sensor is preferable for its applicability at a Level Crossing, a brief survey of each sensor is discussed. Finally, the discussion will conclude with a summary visualised in Table 4.

2.3.1 Obstacle Detection System

2.3.1.1 Inductive Loops

In (Fakhfakh *et al.*, 2011), the author describes Inductive Loops as "probably the most common form of vehicle detection". It is also the earliest proposed solution to detect an object. The Inductive Loops technology has a wire embedded inside the road tarmac in a loop-like shape, which creates an electromagnetic field. The loop acts as an electrical circuit with variable frequencies ranging from 10 kHz to 200 kHz depending on the model used. The object e.g., vehicle when passed through the effective range, induces eddy current and decreases the inductance. The decreased inductance actuates the electronics, which could be used to produce a simplified output. A detailed discussion on Inductive Loops' installation and work principles is given in (A. Klein, K.Mills and R.P. Gibson, 2006). A typical installation of Loop Detectors is given in (Petrov, 2011) as shown below in Figure 5

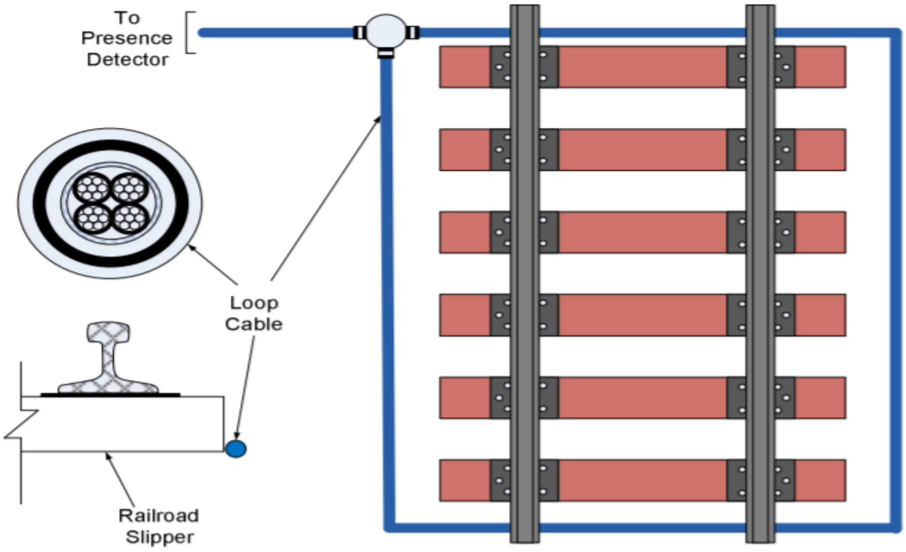


Figure 4A typical installation of Inductive Loops at a Level Crossing.

The Inductive Loops are simple in their functionality but have many disadvantages. The working principle of Inductive Loops depends on inducing the electric circuits which reduce the detectability of objects made from more composite and non-inductive materials. Since most vehicles are made up of composite materials, their detectability is not possible with the given system. Also, in the case of system failures, the embedded system is abandoned and a new system is installed. This can cause significant disruption in areas of high traffic volume and increase the risk of an accident when in proximity to rail infrastructure. Removal and reinstallation of the system are costly and ineffective for the required Risk Model. Extreme weather conditions e.g., heavy rain or fog can adversely affect the system and result in low detectability. Therefore, its inefficiency in poor weather conditions, cost of installation, extensive wear and tear and disruption during maintenance make Inductive Loops an ineffective sensor for its applicability at Level Crossings (Petrov, 2011).

2.3.1.2 Strain Gauge and Piezometers

Another technique used for object detection mentioned in (Darlington, 2017) is using strain gauges or piezometers, which measures the deformation of a material. A strain gauge can be installed inside a crossing area, which can detect the deformation of the crossing deck when a vehicle travels over it. The object's weight causes the decking and attached piezometer to deform, which changes its conductivity and signals the object’s presence above the sensor. Multiple piezometers can be used to interpret the direction of the object. These piezometers are made from rugged, weatherproof semiconductor materials, which usually perform better than Inductive Loops.

However, the installation of these systems in the crossing deck makes it difficult to install and maintain. This further increases the cost and the risk associated with the installation and removal of the system when maintenance is required. Also, the most risk at a Level Crossing is unattended children who are left undetected by many sensors. And since piezometers are calibrated, they cannot detect lighter objects e.g., children. The life expectancy of these products, wear and tear, cost and its inability to detect lighter objects do not justify the implementation of Strain Gauge and Piezometer for its application at a Level Crossing.

2.3.1.3 Stereo Cameras

Another method for obstacle detection at a Level Crossing is having two stereo cameras in one housing as mentioned in (PAVLOVIĆ, PAVLOVIĆ and PAVLOVIĆ, 2016). These two cameras are used to create a 3D representation of the scene when placed at a specific angle to cover an entire area of a Level Crossing. These stereo cameras are integrated with an image-processing unit, where the optical axis, distortions, brightness and other errors are corrected (Ohta, 2005). The basic principle of such systems and how they are installed externally at a Level Crossing is shown below in Figure 6

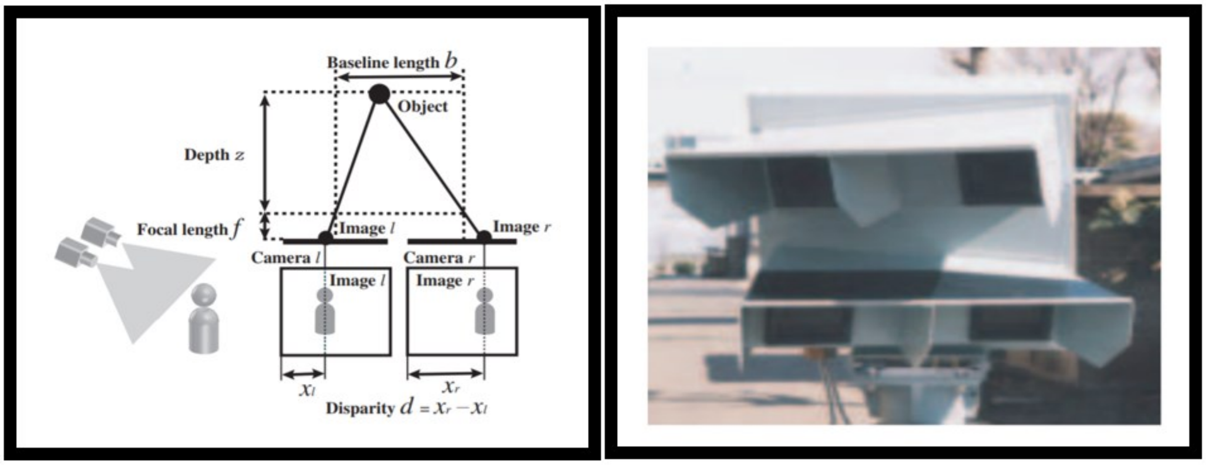


Figure 5a) Principles of the Stereo Camera b) Stereo Camera installed externally at a Level Crossing.

Another significant risk is false positives at a Level Crossing, where a system incorrectly detects an object. As mentioned in (PAVLOVIĆ, PAVLOVIĆ and PAVLOVIĆ, 2016), the stereo camera post-processing techniques can avoid false detection from shadows but traditional post-processing techniques can not differentiate between harmless moving vegetation and a high-risk child's presence. Also, the system has a high installation cost and two cameras as required to cover the entire area, which further increases the cost. The housing box should be regularly cleaned because adverse weather conditions such as fog or heavy rain will disrupt the lens. This increases the maintenance cost and regular interference of a person for maintenance is required. The initial installation cost and regular maintenance and cleaning make it a less favourable choice for a Level Crossing application.

2.3.1.4 Thermal Cameras

The work in (Pavlović *et al.*, 2018) proposed the use of a thermal camera, which creates an image based on the temperature difference. Thermal imaging relies on the phenomenon of black-body radiation produced by all objects with a temperature above absolute zero, therefore, the system is unaffected by the change in weather or varying light conditions (Landsberg, 2014). The work in (FLIR, 2016) shows a thermal image installed for a railway Level Crossing area.

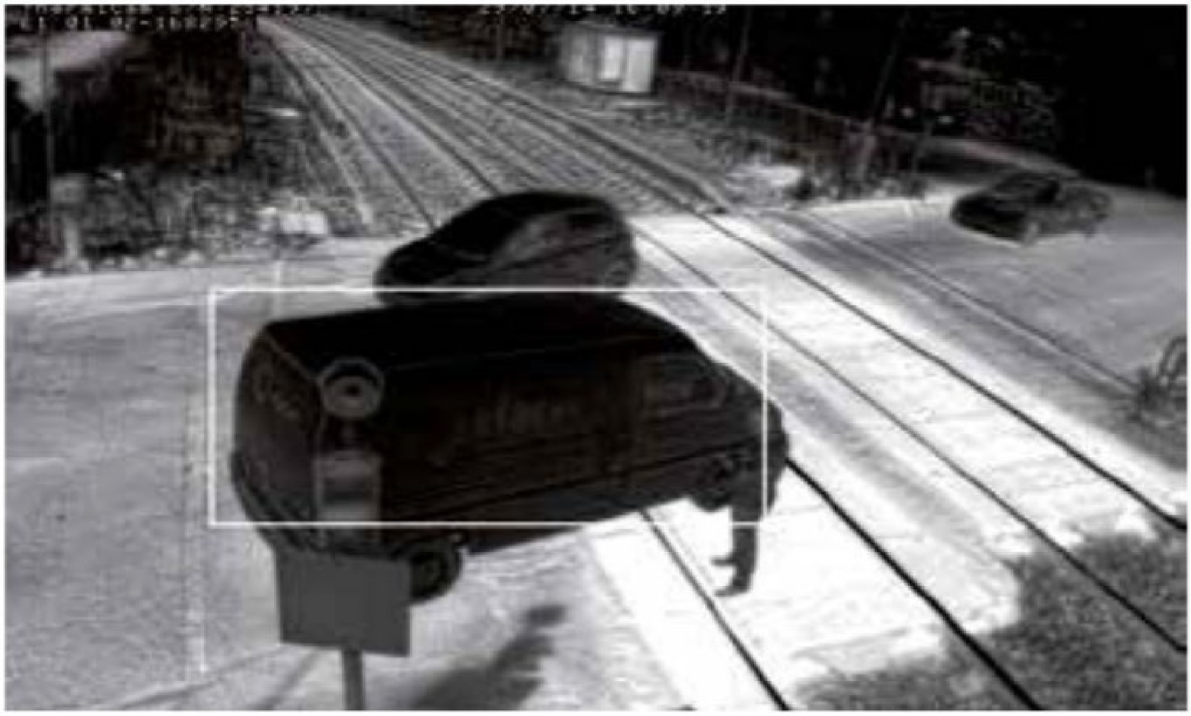


Figure 6Thermal Imaging at a Level Crossing, where Pedestrian and Vehicle is clearly distinguished.

Thermal Imaging provides a high-quality image with easy installation and more options for the lens. As clearly shown in Figure 6, the thermal image can differentiate between a vehicle and a pedestrian easily. The thermal image is not affected by different weather conditions such as heavy rain or fog. However, the system is unable to detect an object which is inert or produce the same radiation as the background scene (FLIR, 2017). Therefore, if the object at a Level Crossing has the same thermal radiation as the background, the thermal image will not be able to detect the presence of the foreground. Also, the thermal image is unable to classify or differentiate between harmful and non-harmful objects. The inability to detect inert objects and classify the objects present at a Level Crossing makes it a less preferable choice for its application.

2.3.1.5 Ultrasonic Detectors

Another technique mentioned in (Fakhfakh *et al.*, 2011; PAVLOVIĆ, PAVLOVIĆ and PAVLOVIĆ, 2016) is ultrasonic detectors and the Optical Beam method. The ultrasonic detectors use the transmission of an ultrasonic pulse and find the difference between transmitted and reflected pulse to detect the presence of a moving or stationary object. The work in (Sato *et al.*, 1998) discusses ultrasonic detectors at the Level Crossings site in detail, where Figure 7 shows how these sensors could be installed at the site.

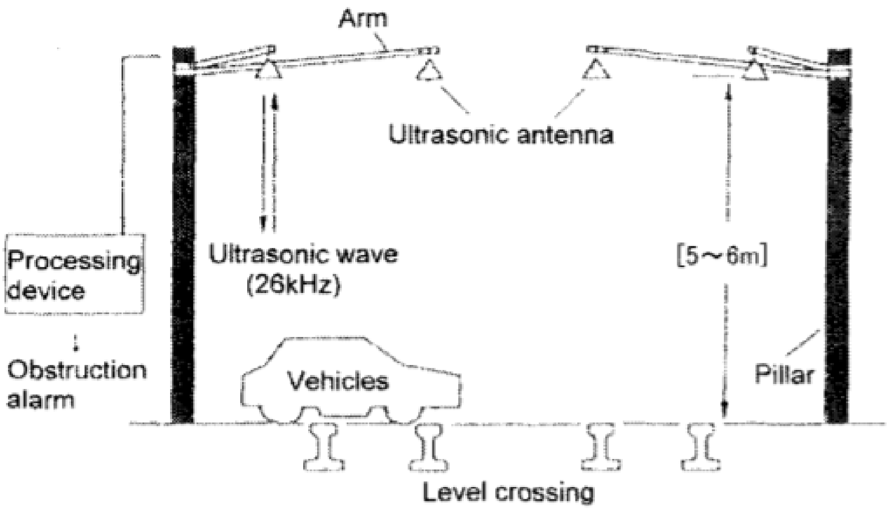


Figure 7A detailed description of how Ultrasonic Detectors is installed at a Level Crossing site.

The effectiveness of an ultrasonic detector relies on its working frequency and size, for instance, the ultrasonic detector with a frequency range of 40 kHz can detect an obstacle about 7 meters away. However, the initial cost of the sensors and their installation cost is high. The effectiveness of these sensors is reduced in adverse weather conditions. The cost and maintenance are big factors to consider for the Level Crossings application. Since these system does not work effectively in all weather conditions and their initial cost is high, the Risk Model does not justify their installation at a Level Crossing (Giannì *et al.*, 2017).

Similar to ultrasonic sensors, the Optical beam works on the same principle. An optical beam is transmitted from one end of a Level Crossing and received on the other end and any interference would detect the presence of an object. Even though the optical beam is easy to replace, the system still has a high installation cost and requires multiple emitters to cover the whole area of the Level Crossing, which further increases the cost. The system cannot detect pedestrians due to its working principle and frequency. Since the pedestrian is at high risk at a Level Crossing as shown earlier and the optical beam is unable to detect them, it makes it a less preferable choice for its application at a Level Crossing (Fakhfakh *et al.*, 2011; Darlington, 2017).

2.3.1.6 Laser Scanners

Another technique for obstacle detection at a Level Crossing as mentioned in (Amaral *et al.*, 2016) is a Laser Scanner, which relies on “the use of dense 3Dpoint clouds produced by a tilting 2D laser scanner”. Figure 8 shows a mechanical Laser Scanner that is used to obtain a 3D image of the scene.

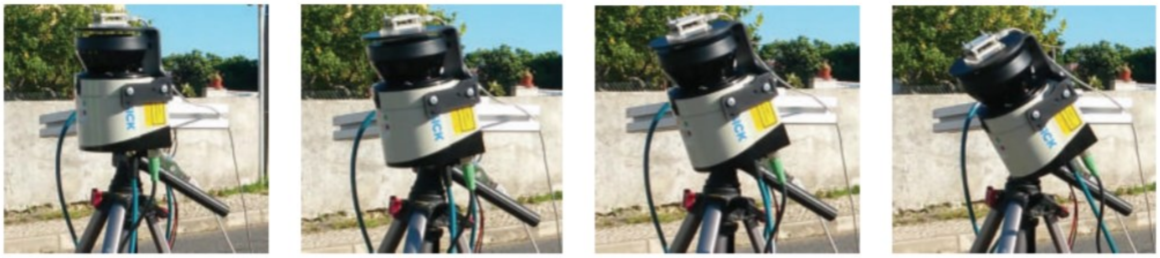


Figure 8A mechanical Laser scanner whose mechanical movement of the system allow the sensor to capture a 3D image.

This system is not dependent on the light conditions and physical properties of objects rather it relies on binocular vision. The working principle of laser is the same as beam or SONAR, where it requires the emitter and receiver.

However, Laser Scanners like any other camera does not perform well in adverse weather conditions. And the cost of installation is further increasing because multiple laser systems are required to cover the entire region of a Level Crossing. In (Kim *et al.*, 2012) the use of “Single Laser Rangefinders” is proposed to avoid the extra cost, where the emitter and receiver are in one unit. But these systems are unable to section areas of a Level Crossing, which is essential to cover an area effectively for its application.

2.3.1.7 LiDAR

Another technique proposed was 3-D Laser Radar in (Hisamitsu, Y., Sekimoto, K., Nagata, K., Uehara, M., & Ota, 2008). It mentions the working principle as “A 3-D laser radar emits a laser pulse to an object, and measures the time that it takes for reflected laser to return to the radar (time-of-flight method) to acquire a distance to that object”.

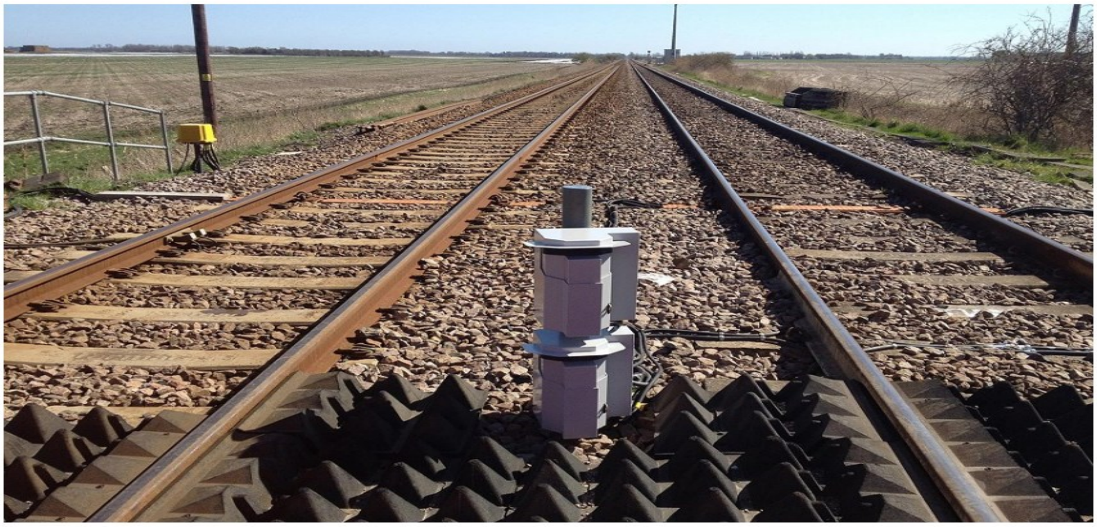


Figure 9LiDAR installed at a Level Crossing site. Picture Courtesy of (Roberts, 2020).

Other work like in (Hsieh *et al.*, 2015; Leddar, 2018) also proposes LiDAR, which works on the same principle of emitting near-infrared light. The reflected light allows the determination of speed, position and direction of an object. They both work on the principle of “Time of Flight (ToF)”, which can be used to determine the distance of an object. The LiDAR is compact, light and robust with high resolution, high range measuring accuracy and high scan rate. This allows the sensing system to detect even smaller objects such as a child. Detection of a small child makes LiDAR one of the preferable choices for its application at a Level Crossing. However, the initial cost and small life cycle of such systems is a limiting factor. Also, the Risk model as discussed earlier requires the data for each category misusing the Level Crossing and it is hard to classify objects from such systems. Therefore, the initial cost and inability to classify objects do not make LiDAR the primary choice of a detector for its application at a Level Crossing.

2.3.1.8 Closed Circuit Television TV (CCTV)

The use of "Camera" has been of much interest to many researchers for the application of object detection at a Level Crossing. The system can detect small objects e.g., children which were impossible compared with many other mentioned sensors. The sensor can classify objects and store the data for the Risk Model as well. With these CCTV systems, it is possible to gather all this information quite easily. The system is easy to install and access to this data is available for post-processing. In (Fakhfakh *et al.*, 2011) it is stated for video sensing system that "an automatic process must be established to only transmit relevant information to the control room and train driver like the presence of an obstacle in the crossing zone at a critical time".

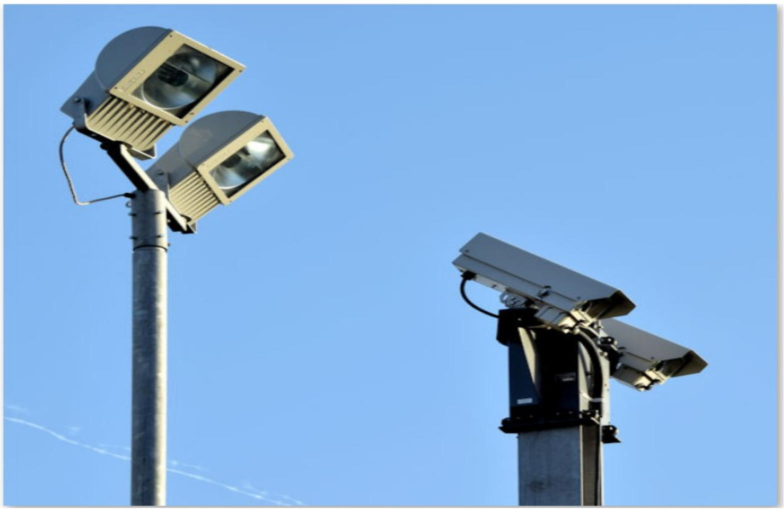


Figure 10A CCTV system installed at a Level Crossing site. Note the flashlights installed along with the CCTV as well to provide enough light for CCTV to operate at low lightning conditions.

The video sensing system is also proposed in (Salmane, Khoudour and Ruichek, 2013) and many different imaging processing techniques are applied to the sensing systems for analysing and automatically detecting objects (Karakose, Akın and Tastimur, 2017). This post-processing of the data allows authorities to automate the whole operational cycle and its ability to replace the manual operator or signaller at a Level Crossing.

The drawback of such a sensing system is that any video system requires a sufficient amount of light to work effectively. During adverse weather conditions such as heavy rain or fog, the detectability of such systems is reduced. However, these issues could be eliminated with the use of flashlights as shown in Figure 10 (Picture Courtesy: (geograph, 2014)). Currently, many Level Crossings within Great Britain has a CCTV system installed, which eliminates or reduces the cost of installation. CCTV system requires low maintenance and have a long life expectancy. The cost of the system, long-life expectancy and easy installation and maintenance make the CCTV system a preferable choice for a Level Crossing application.

Using traditional programming techniques, the ability of video systems to distinguish objects and their properties in any given scene was a difficult task and required extensive manual programming to process and analyse the information acquired from the given pixel. For example, the video system is unable to differentiate between harmless cardboard and a high-risk human child present at a Level Crossing. To distinguish between an inanimate object and a child would require a programmer to define a lot of features that could discriminate them. If extensive programming was not achievable, a manual operator or signaller was required who could classify the child and avoid any harmful situations. The inability to distinguish and classify these objects especially small children posing high-risk to Level Crossings makes the traditional approach inefficient (Valera and A. Velastin, 2005). The usability of such systems is limited by an increase in false positives that is when an object is incorrectly detected. Most false positives are caused by a dynamic background, which resembles a real-world situation. The moving or growing vegetations at most Level Crossings sites would give false positives when used with traditional programming. This is not a failure of the algorithm, but a misunderstanding from its capability to differentiate between harmless and high-risk objects.

The problem lies in the post-processing algorithms, where it is unable to understand the information given in the pixels, therefore, the upgrading is required in the algorithm, not the sensing system. For this purpose, the proposed work will introduce Deep Learning; an algorithm that can learn features and representations automatically without manual programming essential for classifying and distinguishing objects. With Deep Learning techniques, the CCTV system can detect and classify objects, also gather enough data for the Risk Model required for a Level Crossing application.

2.3.1.9 RADAR

RADAR is another area of interest for many researchers and is mostly used obstacle detectors at a Level Crossing within the UK. The working principle of RADAR is the same as emission and receiving of a signal e.g., radio waves, the received echo suggests the presence of an obstacle. Different types of RADAR are available whose working principle remains the same but differ in their post-processing to obtain information such as distance or velocity. For example, the pulse RADAR uses the Time-of-Flight method and Frequency modulated RADAR determines the distance from the difference in the emitted and received frequency of an echo (Horne *et al.*, 2016). Figure 11 (Picture Courtesy (Dent and Marinov, 2017)) shows a RADAR installed at a Level Crossing site in East Sussex.

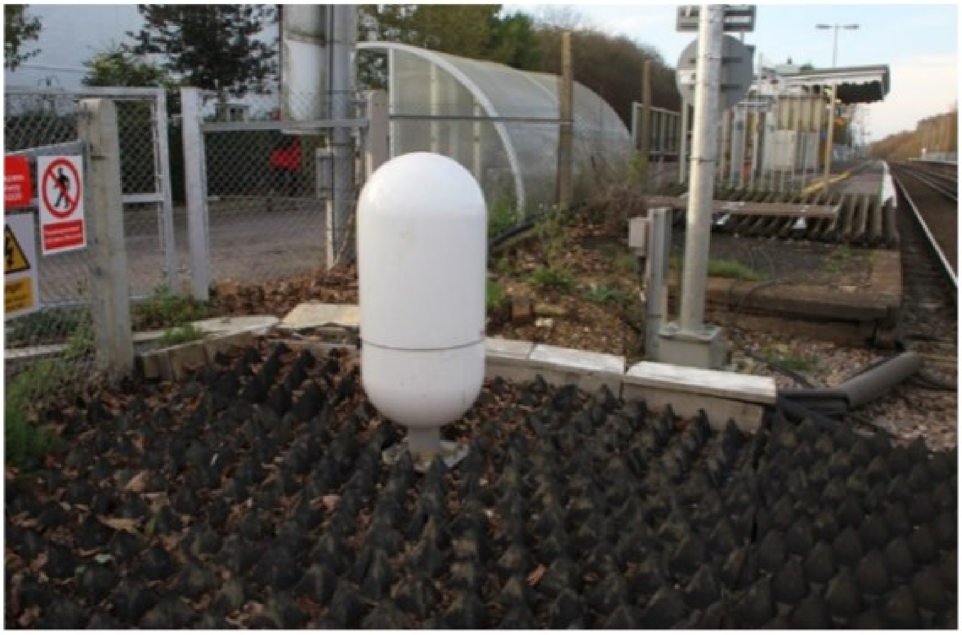


Figure 11A RADAR sensor installed at a Level Crossing site in East Sussex, UK.

In (Hilleary and John, 2011) the implementation of RADAR in four-quadrant gate Level Crossing is discussed in detail. It discusses the use of the Timed and Dynamic method of operation, where the Timed method delay the closing of the barrier if some vehicle is present and in Dynamic the operation depends on the presence of the vehicle within the "Minimum Track Clearance Distance (MTCD)." Also, in (Addabbo *et al.*, 2016), the RADAR is proposed as the main sensing system to detect obstacles with three safety configurations. Traditionally, the RADAR is used to detect an object and start the automation process of a Level Crossing operational cycle. Although, it works well and can detect even small objects present e.g., children. However, the RADAR itself will not be able to distinguish a high-risk child from harmless cardboard. This would require some sort of post-processing techniques. With traditional post-processing techniques, the RADAR can acquire velocity, distance or direction of an object and from such properties, it can categorise objects. But the categorisation from such properties would be quite inefficient. To overcome such a problem, the proposed work introduces the use of Deep Learning techniques (Mun, Kim and Lee, 2018). The most common application for RADAR with its integration with Deep Learning is using SAR (Synthetic Aperture Radar) Imaging, whereas, the proposed work will use micro-doppler signals acquired from the RADAR as an input. From micro-doppler signals, the model will learn representation itself without the intervention of a manual programmer using a Convolutional Neural Network. The trained model can categorise these objects more efficiently and give reliable results because the RADAR produces distinct signals for each small movement of an object, which is used to generate micro-doppler signals.

Since most of the Level Crossing sites within Great Britain has RADAR installed along with the CCTV system, the cost of installation will be significantly reduced or eliminated. Also, Network Rail is planning to replace the old MK1 RADAR with pole-mounted RADAR scanners. The new scanners will replace the old RADAR and LiDAR at Level Crossings across Great Britain. The new RADAR will cover an entire region of a Level

Crossing and provide micro-doppler signals for the proposed model to learn representation, which is used to detect and classify an object (Network Rail, 2020c).

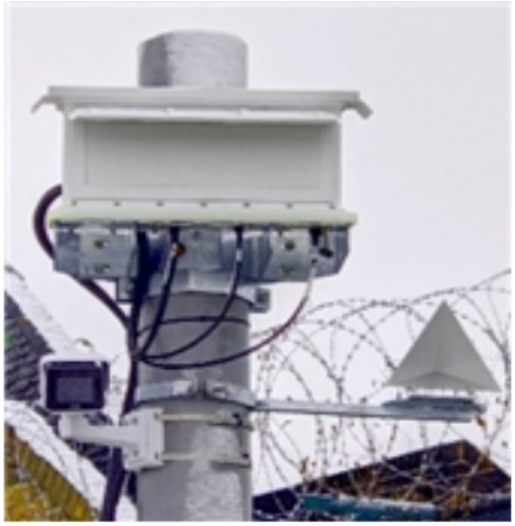


Figure 12Second Generation Obstacle Detector-OD2. The OD2 is pole mounted RADAR scanner, which is able to replace old RADAR and LiDAR at a Level Crossing site.

The low maintenance and high life expectancy of approximately 10 years further reduce the cost of the sensing system. Most of these RADAR works in the range of Gigahertz wavelength e.g., 24GHz, which makes the system more robust in different environmental conditions. It can easily detect small and stationary objects at a Level Crossing, which pose the most threat to the crossing's safety (Govoni *et al.*, 2015). These characteristics of RADAR makes it a primary choice of Obstacle Detector for a Level Crossing application.

2.3.2 Sensor Fusion

Using a traditional approach, a single type of sensor cannot provide sufficient information about the object, which is required for its application at a Level Crossing. For example, the functionality of video camera-based systems is limited by the availability of light and RADAR is unable to classify between harmless cardboard and the presence of a high-risk child. Given the limitations of using a single type of sensor, the use of two or more sensors is a common practice adopted by many different organizations for different applications. The fusion of sensors adds another layer of resilience and therefore overcome the shortcomings of a single sensor. The interlocking system discussed earlier with the "2oo2" approach requires two processing channels e.g., two sensors working independently for one purpose. The proposed work will use CCTV and RADAR as two mutually independent processing channels to give an output, which ultimately automates the operational cycle or alarm the train driver of a high-risk situation using a communication network e.g., GSM-R. Some other common sensor fusion approaches are discussed here as well.

2.3.2.1 Camera & LiDAR

In (Manduchi *et al.*, 2005) the fusion of colour Cameras and LiDAR is suggested, where colour Cameras can recognize distinctive objects because of their colours and texture using different post-processing algorithms and LiDAR can discriminate such objects. The underlying issue with colour cameras is their inherent ambiguity due to the varying reflectivity of an object such as dry grass and soil. Also, the effect of the illumination spectrum on the perceived colour and the chromatic shift due to the atmosphere pose additional problems. The dynamic environment of a Level Crossing e.g., growing vegetation reduces the effectiveness of a Camera system. To overcome the mentioned problems, the LiDAR is proposed which can discriminate the moving vegetation from other smooth surfaces e.g., rocks or tree trunks. The LiDAR can eliminate the shadow effects as well as complimenting the Camera system more effectively.

Two Colour Cameras are required to cover an entire area of a Level Crossing site; hence, the installation cost is high. The LiDAR's cost of installation and maintenance is high as well. The high cost of installation and maintenance does not justify the implementation of such a system at a Level Crossing site across Great Britain. Hence, the implementation of such systems is not preferable for a Level Crossing application.

2.3.2.2 Multi-Beam & Sector Scanning Sonars

In (Ganesan, Chitre and Brekke, 2016) the fusion of Multi-beam and Sector Scanning Sonars is proposed. Sector Scanners provides a set of scan lines by mechanically moving the unit to obtain an image; hence some image compensation would be required to produce an accurate image. To avoid this limitation, the multi-beam is proposed which will scan the entire area reducing the need for stabilisation. The proposed system works in underwater situations and has not been tried in an outdoor environment.

Neither of these sensors can differentiate objects, hence, the fusion of such sensors for a Level Crossing will increase the number of false alarms. These false alarms will pose a high risk for a Level Crossing application and pose a threat to the road user and train operator. Some traditional processing techniques may be used to segment an image and extract features for differentiation but manual programming will not be an effective solution as discussed earlier. The fusion of such sensors would suggest the replacement of every sensor installed at each Level Crossing site and replace them with Multi-Beam and Sonars. The approach will not justify the cost and the limitations of such sensors will further make it the least preferable choice for its applicability at a Level Crossing.

2.3.2.3 RADAR & LiDAR

LiDAR and RADAR are the most commonly used sensor at a Level Crossing site, their fusion is proposed in (Birtles, 2017). RADAR is used as a primary obstacle detector to cover an entire area of a Level Crossing using multiple sensors. The data is post-processed to deduce information about an object such as speed and distance. The RADAR, however, may not be able to detect small density objects. To avoid such limitations, the use of LiDAR is proposed, which can detect and obtain a detailed map e.g., a 3D scene of the scenario. The LiDAR can detect small density objects, which are missed from the RADAR.

However, the LiDAR is normally within a glass unit; therefore, the presence of dust or small water droplets affects the functionality of the system. To overcome such a problem regular maintenance is required, which further increases the cost of the system. The RADAR from post-processing techniques may acquire information about the objects e.g., speed, distance or direction but this information is not sufficient to classify objects effectively for the Risk Model. LiDAR and old MK1 RADAR is being replaced by new OD2 RADAR across Great Britain. Hence, the fusion of such sensors may not be the best possible solution given the current sensor configuration at a Level Crossing site.

The sensor fusion at a Level Crossing should be complimented with sensors that are already installed at a Level Crossing site to reduce the cost and justify its implementation. As mentioned, the sensors currently installed at Level Crossings are CCTV, RADAR and a few sites LiDAR. But new generation obstacle detector is replacing the RADAR and LiDAR with just one pole-mounted RADAR. The fusion of CCTV and RADAR is proposed in this work and is discussed later with the integration of Deep Learning techniques.

2.3.3 Summary

At this point, it is necessary to provide a summary of all the technologies mentioned and discussed earlier in Table 4, where scores are used for each technology. The 1 represents the high effectiveness and 5 represents the least effective for its parameters e.g., Cost, Maintenance etc. for the given sensor. As mentioned in Table 4, the CCTV and RADAR have low cost and maintenance with low false alarms. The fusion of CCTV and RADAR would add a two-layer resilient system, which demonstrates the "2oo2" approach for the Interlocking system with SIL of 3-4 for its applicability at a Level Crossing. Both of these sensors would use Deep Learning techniques to learn features and representations essential for classification and detection. The classification of objects would allow the Risk Model to collect sufficient data to analyse it and take safety precautionary measures against the particular category which mostly misuses the particular type of a Level Crossing.

	Equipment cost	**Maintenance**	**Product Life Cycle**	**False Positive**	**Impact on Weather**
Inductive Loops	5	5	4	4	4
Strain Gauge/Piezometers	5	5	4	4	3
CCTV	2	3	2	3	3
Stereo Camera	3	4	2	3	3
Thermal Camera	4	2	2	2	2
SONAR	2	4	3	3	2
Millimetre-Wavelength Beam Interference	4	4	1	3	4
Laser Range Finder Beam (LRBF)	3	4	2	5	4
Light Detection and Ranging (LiDAR)	5	4	2	2	2
RADAR	3	2	1	1	2

Table 4A brief overview of each discussed sensor. The 5 represents the worst case and 1 being the best choice for its applicability. The given summary should help the reader select the most appropriate sensor for their given application.

It is evident from the above discussion that the best possible sensors to be utilized for a Level Crossing application is CCTV and RADAR. The discussion takes account of the installation and maintenance cost, the reliability and effectiveness of a Level Crossing application and how these sensors can effectively provide sufficient data for the Risk Model, which is essential for the long-term operational cycle of the system. The proposed work focuses on Great Britain sites, where CCTV and RADAR are primary Obstacle Detectors already installed for a Level Crossing application. This will significantly reduce the cost of installation and justify the implementation of the proposed work through the Risk Model. The system's life-cycle and low maintenance further reduce the cost, which makes these two sensors a preferable choice for its application.

2.3.4 Obstacle Detection Algorithms

The data received by any of the aforementioned sensors require some sort of post-processing before it can be used effectively in any given application. The post-processing techniques may include image-processing, feature extraction, image segmentation or foreground detection. The ultimate purpose of such an algorithm is to detect the presence of an object at a Level Crossing, which will either start the automated process of closing the barrier or warn the train driver of any threats. Traditional Algorithms are used to detect the foreground from the background scene, and since most of the Level Crossing sites has CCTV installed these traditional programming's use pixel values to detect a foreground. The problem with such an approach is many depending on the algorithm used (as discussed further in this work). However, the general issue with such an approach is a dynamic background because almost all of these traditional algorithms would somehow compare the background pixel with the new frame. The site at a Level Crossing is dynamic, where vegetation grows and moves with the wind. This means that the algorithm should adjust with the new changing background scene constantly and this makes the traditional approach less effective. However, the proposed technique in this work learns representation from the data regardless of its shape, size, orientation and position in an image. The ability of the new proposed technology to learn representations rather than depending on background pixels makes it a preferable choice for its applicability for a Level Crossing. Before discussing the newly proposed technique called Deep Learning, it is reasonable to discuss those traditional algorithms currently used to detect foreground from the background scene.

2.3.4.1 Single Gaussian (SG)

The work proposed in (Patel and Tank, 2015) introduces the algorithm that uses the "Single Gaussian-SG", which fits the Gaussian probability density function on the last 'n' pixel's value. To avoid fitting the mean and variance are calculated using the "density function" for every new frame. To detect any foreground, the given value is compared with the threshold value. The work in (Wren *et al.*, 1997) discusses the use of Hue-Saturation-Value (HSV) compared with the traditional use of Color Scheme (Red, Green and Blue-RGB), which is more robust to illumination changes and partial camouflage.

2.3.4.2 Mixture of Gaussian (MOG)

For outdoor applications where the background is a dynamic environment, the work in (Stauffer and Grimson, 1999) proposed the use of "Mixture of Gaussian-MOG". Each pixel is modelled using "Mixture of Gaussian" and each model uses a one-line approximation to update the model. The optimal number of Gaussian filters is updated according to the given environment. Another work in (Allili, Bouguila and Ziou, 2007) models the background using three Gaussian filters representing the road, the vehicle and the shadows, where the intensity of each pixel is used for this categorization; the darkest component is labelled as a shadow, from the remaining two, the one with the highest variation is classified as a vehicle (obstacle) while the last one is classified as a road. The Expectation-Maximization (EM) and the K-Mean algorithms are two of the most commonly used approaches to model the background and detect the foreground. The number of distributions (K) must be predetermined, which can be a limiting factor for this model. The Mixture of Gaussian is an effective method for backgrounds with small illumination changes; also, the model does not require storage of important input data in the running process.

2.3.4.3 Mixture of General Gaussian (MOGG)

The problem with the “Mixture of Gaussian” model is the noise and clutter in a given environment, therefore, the work in (Allili, Bouguila and Ziou, 2007) proposed the use of “Mixture of General Gaussian-MOGG”. A finite mixture model of the general Gaussian filters is used for robust segmentation and data modelling in the presence of noise and outliers. The MOGG model is more flexible to adapt the shape of data, and less sensitive to over-fitting compared with the MOG model. Each pixel is characterised by its intensity in the RGB Colour Space. The optimal number of Gaussian filters is computed at each time “t” by minimising the criterion of Minimum Message Length (MML). If the number of Gaussian filters at time “t+1” is smaller than at time “t”, the parameters are updated similarly to a Mixture of Gaussian. A Mixture of General Gaussian approach shows better results than a Mixture of Gaussian in environments with shadows (Allili, Bouguila and Ziou, 2007). The work in (Friedman and Russell, 1996) and (Allili, Bouguila and Ziou, 2007) uses the Expectation-Maximisation (EM) and the K-Mean methods for MOGG models that adapt to the data more and are less sensitive to over-fitting.

The problem with these mentioned Gaussian models is their inability to update the dynamic background model. The Gaussian model works where there is minimal illumination variation and the background model does not change immensely, however, the Level Crossing site is quite dynamic and complex with constant change in illumination and background scene. Also, the Gaussian model recovers slowly from false positives, which would delay the process and pose a significant risk to Road and Rail users in case of failure. The system at a Level Crossing should work in real-time with minimum recovery time. Hence, the Gaussian Models are not a preferable choice for their applicability for Level Crossings.

2.3.4.4 Kernel Density Estimation

The work in (Piccardi, 2004; Patel and Tank, 2015) introduces another technique that uses the probability density function using the K estimator on a recent sample (N) of intensity values. From the normal Gaussian function, the background model is obtained using Kernel Estimation- KE function. The foreground detection is possible using a threshold value, which is also used to obtain two more models; the Short-Term background model and the Long-Term background model. The Short-Term model eliminates persistent false positives, whereas, the Long-Term model reduces the number of false-positive detection. The use of these two updated models makes the system more effective and reliable for dynamic background environments compared with other traditional techniques (Elgammal, Harwood and Davis, 2000).

2.3.4.5 Subspace Learning

Another algorithm is proposed in (Leonardis, 2002), which uses “Subspace Learning-SL”. Subspace Learning is divided into two main stages; Reconstructive and Discriminative. The Reconstructive Phase learns as much information as possible from the data through unsupervised learning and updates the data in increments, which is beneficial for real-time applications. The Discriminative Phase requires supervision to separate the data using a linear transformation, which is used for classification. The training requires the data in batch; hence, the data should be available in advance which is a downside. Also, the model is computationally extensive and creates a lower-dimensional classification space. Some of these limitations are overcome by different techniques as discussed.

2.3.4.6 Subspace Learning using Principal Component Analysis

This model deals with the illumination changes by considering the spatial features. Each background pixel is modelled using the Eigen background model. This model consists of an “N” number of images for which the mean and covariance matrix are calculated. To reduce the dimensionality of space, only a certain number e.g., “M” eigenvectors are considered in Principal Component Analysis. The “M” eigenvectors correspond to the largest eigenvalue in the background matrix. The Input and background images are compared to a threshold value to detect the foreground (Bouwmans, 2011).

2.3.4.7 Subspace Learning using Incremental Non-Negative Matrix Factorization

The INMF matrix decomposes the data matrix into the mixing matrix and encoding matrix. INMF aims to find an approximate factorization that minimizes the reconstruction error (Bucak, Gunsel and Gursoy, 2007). The most commonly used error function is error squared due to its simplicity and effectiveness. Background initialization is made using “N” training frames. The matrices are updated incrementally. Foreground detection is made by thresholding the residual error which corresponds to the deviation between the background model and projection of the current frame onto the background model (Li *et al.*, 2003).

2.3.4.8 Subspace Learning using Incremental Rank Tensor

The IRT considers the spatial information to detect a foreground from the background scene. This online algorithm constructs a low-order tensor eigenspace model in which the sample mean and Eigen basis are updated. Once the model is computed, foreground detection is possible using the threshold value. This method has shown more robustness than the PCA method (Hu *et al.*, 2013)

Subspace Learning can retain as much information as possible and using linear transformation can classify the objects, however, the use of threshold value means it relies on the background pixels. Given the dynamic environment at a Level Crossing, the background scene will change constantly, hence the need to update the background is evident. Many different sub-techniques are available for Subspace Learning some of which are discussed earlier. These sub-techniques rely on background pixel values somehow to approximate the presence of a foreground, which again is not an effective method for a Level Crossing application.

2.3.4.9 Support Vector Machine

Some algorithms are used to separate the data used for classification. The work proposed in (Bouwmans, 2009) is “Support Vector Machine-SVM”. The SVM utilises a hyperplane in high dimensional space to separate the data by minimising the margin between the hyperplane to the data. The output is divided into two sets for unbiased training; 80% for SVM training and 20% for two parametric minimisations. If the probabilistic output is greater than the distance of the hyperplane to the data, the output is considered as either a new moving foreground or a newly updated background.

Support Vector Machine is the most basic Machine Learning algorithm used for classification and regression problems. However, the SVM is used for linear classification where features are manually selected before using the classifier. Also, the model does not work in complex situations like Level Crossings. Comparatively, the Convolutional Neural Network (CNN) trained via Deep Learning techniques is used for visual recognition and image classification idle for its applicability at a Level Crossing. A more dynamic and multi-class classification is possible from CNN using multiple layers, which is not possible from SVM (Sharma, 2018).

2.3.4.10 Support Vector Regression

The model proposed in (Wang, Bebis and Miller, 2006) uses the "Support Vector Regression- SVR", which models each pixel as a function of its intensity. The pixel values of each frame contain two outputs, one corresponds to the intensity of the pixel and the other value represents the confidence of that pixel belonging to the background. Once trained, the model uses a linear regression function where the output of SVR represents the confidence of each pixel belonging to the background. The SVR model keeps updating the background scene by labelling the pixel as background if the confidence lies within the threshold range. The model is constantly updated using an online SVR learning algorithm.

But like other discussed traditional approaches, the SVR relies on the pixel values for background and foreground detection. This approach would work where the background is static or with minimal changes. However, the Level Crossings requires a model, which is efficient and reliable for the dynamic environment and updates the model in real-time with minimal recovery time. This is not possible with any algorithm which relies on the pixel values; hence it is not preferable for its application at a Level Crossing.

2.3.4.11 Support Vector Data Description

The work in (Tax and Duin, 2004) plots the background model using the Support Vector Data Description (SVDD) method. The SVDD method is used in videos with a quasi-stationary background, where data domain descriptions concern the characteristics of the dataset. The boundary is used to detect any outliers present, where the simplest boundary is represented by a hypersphere with centre "α" and radius "R". Optimisation techniques are used to minimise the volume of the sphere while keeping the whole training sample within. Once optimised, the new frame is used to measure the distance from the centre and compared with the value R, if the new value lies outside the given R-value it is considered as a foreground.

The use of threshold values and probability functions limits the functionality of the proposed algorithm. To overcome this issue, the work in (Tax and Duin, 2004) proposed the use of "Support Vector Data Description-SVDD", which uses the boundary feature. However, the optimisation is computationally intensive and during maintenance, all SVDD must be recomputed, which is not an efficient approach for real-time application at a Level Crossing.

2.3.4.12 Temporal Median Filter or Temporal Differencing

Another technique mentioned in (Chinmayi *et al.*, 2017) is the use of "Temporal Median Filter-TMF". The TMF uses the median value of the last frame to model the background, which increases the stability of the background. However, the TMF cannot accommodate the model in the rigorous statistical description and does not provide a deviation measure for the adoption of the threshold value, which means it cannot adapt to the dynamic environment, which makes TMF the least favourable choice for its application at a Level Crossing.

The work in (Lipton, Fujiyoshi and Patil, 1998) proposed the use of Temporal differencing and Image Template Matching to detect and track the objects in a video sequence. The method is robust to background clutter and a slight change in the object's presence would not significantly affect the performance.

2.3.4.13 Co-occurrence of Image Variation

The work in (Seki *et al.*, 2003) proposed the use of "Co-occurrence of Image Variation", which works best if the background object belongs to the same object with the least variation. These classification algorithms make use of the concept that neighbouring pixels belonging to the same object would have the same variation over time. The method can be divided into two main steps; the Learning Phase and Classification Phase. The Learning Phase calculates the eigenvector value, which reduces the dimensions of image variation. In the classification phase, a block with its current input value is computed along with its corresponding Eigen image variation value. The nearest neighbouring block is expressed as an interpolation of the calculated value. After applying the interpolation coefficients to the current block and neighbouring block, a new value is obtained. Likewise, the probability of 8 neighbouring blocks is calculated and if the calculated and new values are close then it is assumed that they both belong to the background model

However, if the objects are distant, the algorithm will give many false positives since it only works with close neighbouring blocks. Level Crossing is a dynamic environment with distinct objects at varying positions, which makes the proposed algorithm a very poor choice for its applicability at a Level Crossing.

2.3.4.14 Kalman Filter

Since most of these traditional approaches rely on some "threshold value", which could give false alarms for the dynamic backgrounds. To avoid such an issue and false alarms, dynamic thresholding is proposed (García *et al.*, 2010). In "dynamic thresholding" every correlation output is estimated by first-order polynomial interpolation, where a Kalman Filter is used to estimate the system output and obtain a dynamic threshold.

To utilise the Kalman Filter, it is assumed that noise signals are of a "zero-mean" value. The system output is estimated and the threshold at the "k" frame is half of the estimation at "k-1". Another filter called "H∞ filter" is used as minimax filtering, which tries to minimise the maximum estimated error, which is considered good practice for the worst-case scenario during its application. The "H∞ filter" requires more time compared with the Kalman filter to perform adequately and shows better results. Although, it works well for object tracking using dynamic thresholding but fails to classify objects, which is essential for a Level Crossing application.

2.3.4.15 Principal Component Analysis

Principal Component Analysis (PCA) is another method to classify, which is divided into two phases; an offline training phase and an online detection phase. In the training phase, varying operational conditions are considered with a section of track, which is free from objects. The data is used to obtain a transformation matrix "U" between the original space and transformed one or vice versa. In the online phase, the received measurement from the processing unit is compared with the original space. The error between these two measurements can either be too large or small in magnitude depending on the similarity that exists between these two spaces. If the error is larger than the pre-defined threshold value, it suggests the presence of a foreground (García *et al.*, 2010). A more generalized version of PCA is available; Independent Component Analysis (ICA) is discussed below.

2.3.4.16 Independent Component Analysis

Independent Component Analysis (ICA), which provides a clearer distinction between the foreground and a background object. The ICA is less sensitive to noise and can detect small and stationary objects without any prior knowledge of objects using data variables. For example, in an outdoor environment, ICA uses

Red-Green-Blue (RGB) channel colour scheme, where a channel with the highest signal/noise ratio is used for the motion segmentation process. The histogram of the output channel is used to calculate the threshold value.

The ICA algorithm is used to generate a demixing matrix. The matrix is formed by random background and the image with an object in the foreground. The foreground is independent of time and space. The two clear images are taken as an input to ICA and the data matrix is computed. The inverse of the mixing matrix is the demixing matrix generated by the Fast-ICA algorithm. The Fast-ICA algorithm is a fixed-point iterative scheme maximising the non-Gaussianity as a measure of statistical independence. It tries to find the independent component by estimating the negentropy. The algorithm iteratively searches for weight in a set matrix of a Neural Network from a dataset that properly separates the data signals in the mixture into its independent components. More detail on ICA is given in (Masaki Yamazaki, Gang Xu, 2006). Both of these approaches rely on a threshold value compared with a calculated value using a set of data or static pixel values. The dynamic environment at a Level Crossing needs a more robust and effective approach for calculating the foreground.

Some other works proposed the fusion of two techniques such “K-Mean clustering” algorithm on a modified “Lucas-Kanade” method (Šilar and Dobrovolný, 2013), “Lucas-Kanade” technique with “Haris Corner Points” in (Salmane, Khoudour and Ruichek, 2016) or “Histogram of Oriented Gradients-(HoG)” algorithm with a Support Vector Machine-SVM in (Junghans *et al.*, 2016). All these approaches combine two different traditional techniques to give a more reliable output, which is similar to the “2oo2” approach for the Interlocking system. However, the traditional approach relies on a similar methodology to calculate or detect a foreground, which is to use static background pixels and use some threshold value to detect the foreground. Since both processing channels are using the same techniques, both will likely give false alarms at the same time which defy the main purpose of the “2oo2” Interlocking system. Therefore, the proposed algorithms should not depend on background pixels to detect the presence of foreground and they should not use the same processing channel to give an output for an effective “2oo2” approach for a fail-safe mechanism at a Level Crossing.

2.3.5 Summary

The above discussion is sufficient to suggest that the use of traditional algorithms is not efficient for their applicability at a Level Crossing. The traditional algorithms rely on calculating the background pixel values to model the background, which then confirms the presence of the foreground using a pre-defined threshold value. This method does not depend on the object features or recognize objects from their specific properties, rather on the variation within the background scene. This further suggests that any variation in background scene due to change in lighting conditions or dynamic environment e.g., growing or moving vegetations would give false positives. These false positives would pose a significant threat to road or rail users. Also, the recovery time for most of these mentioned algorithms is slow, which means more chances of failure, which is not preferable for its application at a Level Crossing. Also, the aforementioned algorithms are not applicable to gather data for the Risk Model, which could categories the obstacle for their misuse. The gathered data would help relevant authorities to take precautionary measures to make Level Crossings safer and reliable. Hence, the Deep Learning technique is proposed that does not depend on pixel values, rather it learns representations and features from the training data. Once trained, the model can detect and classify each category present at a Level Crossing site using the features it learned during the training process. The Deep Learning technique will allow the system to make real-time predictions and in the case of false positives, the system would recovery time is quick e.g., prediction at a rate of 15fps which further improves the efficiency of the system. The proposed work would therefore provide a fail-safe system of SIL level 4 and gather enough data for the Risk Model without making the system complex.

2.3.6 Deep Learning

The discussed traditional algorithms work on a principle of either subtracting the foreground pixel with the background pixel and detect the presence of an object using some threshold value or some supervised programming is required to train the data for classification. These limitations affect the functionality of its application at a Level Crossing because it is a dynamic environment and extensive expertise and manual programming are required to classify certain objects with specific properties. To overcome these problems, the work in (LeCun, Bengio and Hinton, 2015) proposed the use of the Deep Learning technique. Deep Learning is a subset of Machine Learning, which can automatically learn the representation from the images and classify the given categories without any supervision. It consists of multiple processing layers and can learn representation using backpropagation algorithms. With enough representations, very complex functions are learned used for different applications such as classification and detection.

The Deep Learning model learns features for the general-purpose avoiding the need for human engineering, for example, the classification model learns small features like edges, orientation and location to more complex representations such as motifs or parts of objects with assembled features. From such learned models, it can detect and classify objects very effectively. The model does not require any supervision or regular increment of data to update the existing system. Another advantage of Deep Learning is its ability to integrate with other existing sensing systems such as RADAR, LiDAR and Video System. For example, the Deep Learning model can classify and detect objects in real-time using the live stream of video from the Video Sensing system. The training on such models requires high computational power, however, once trained the network can work in real-time with an even 15fps rate of processing power. The work in (Colangelo *et al.*, 2018; Wang, Choi and Gopalakrishnan, 2018; Matlab, 2020b) mentions details about the training and computational power required.

Some work has discussed the integration of Artificial Intelligence-AI with different sensing systems. For example, the work in (Manikandan R and S, 2017) discussed the integration of Video Cameras with the use of AI and the work in (Shetty *et al.*, 2019) discussed the use of TensorFlow (an open-source library for Machine Learning) to detect objects. Another work discussing the integration of the Vision system with AI is mentioned in (Pu, Chen and Lee, 2014). These works and the mentioned discussion on each given algorithm suggest the importance of Deep Learning techniques, which can automate the process of learning and classifying the objects in real-time. Detail discussion on the proposed work is presented later in the work.

2.4 RISK ASSESSMENT

The system's availability, safety and cost-effectiveness are the most important aspect of any applied railway system, therefore, the need for a system that should continuously improve the safety, availability and reliability of the railway system is inevitable. The railway industry has considered and introduced many different techniques and methodologies to improve the safety of the infrastructure within the Railway Industry. For this particular work, the RAMS (Reliability Availability Maintainability and Safety) management system is preferred, which is inherited into the Railway systems through some engineering process to achieve efficient and reliable traffic service within railway e.g., Level Crossing operational cycle. Different organizations have successfully utilized the RAMS management system for different applications, however, at the domestic level e.g., Level Crossing the authorities are reluctant to introduce RAMS (JU *et al.*, 2011). This is because a systematic approach for RAMS management systems is not available for the Railway system's engineering, concepts, methods, techniques and tools (Vintr and Vintr, 2007). Lack of consideration in low operational cycles e.g., Level Crossing can influence the quality of rail transport such as delay in trains, increase in cost and serious damage to the environment and human users. Therefore, the present work will discuss the systematic RAMS management system in considerable detail for authorities to analyses and maintain the safety of the existing and new systems at a Level Crossing.

2.4.1 RAMS Management

The RAMS systems are an enlarged engineering discipline, which originated from the concept of safety and reliability (Ebeling, 2004; An, Baker and Zeng., 2005). This was introduced to assess the product failure and human error, where the first assessment techniques were established in the 1940s; "Failure Mode and Effect Analysis (FMEA)". The FMEA was further evolved to "FMECA", where the aspect of "Criticality" was introduced as well (Nicholls, 2005). The RAMS management system has been used by the US railway industry since the 1980s and by European industry since the 1990s. These industries applied RAMS to achieve safety, availability and cost-effectiveness in the management aspect of the system's long-term operation. Firstly, the European Committee for Electrical Standardization (CENELEC) standardized the RAMS management system. This was adapted as an International standard (IEC 62278) along with its family standards: BS EN 50128 (2009) and BS EN 50129 (2003) and now these standards have been used in global railway projects.

BS EN 50126-1 (1999) and the work in (Avizienis, Laprie and Randell, 2001) has defined the RAMS management system into three aspects

1. The definition of four characteristics of RAMS to achieve its requirement and operational contexts. These four characteristics are Reliability, Availability, Maintainability and Safety.
2. Assessment and control of all potential threats which adversely affect the achievements of RAMS requirements
3. Provision of the means to achieve the RAMS requirement for the system.

Reliability refers to the ability of the system to perform its specific function over time without any defined failure, whereas maintainability is the ease of maintenance within the structure of the system. Availability refers to the ability of the system to operate whenever required by the operator and safety refers to the freedom from unacceptable risks with regards to the operation, maintenance, person, environment and equipment. The reliability and maintainability are determined by the system, operation availability and product design, whereas safety and availability are maintained by the system availability. (Milutinović and Lučanin, 2005). Since the proposed work focuses on the safety of the system, it is essential to discuss System RAMS management.

System RAMS management assess and control all potential hazards to the system. The potential hazards such as faults, errors and failures may arise from sources like operation, system and maintainability, which is an important aspect of the RAMS management process (Ann, Rausand and Utne, 2009)o. This allows the management system to provide means of prevention, tolerance, removal and forecasting any faults.

The standard EN BS 50126-1 (1999) mentions that the Railway Industry uses precaution which is a combination of prevention and protection to minimize the possibility of impairment. In summary, the System RAMS management system defines the requirements, access, control threats, plan and implement RAMS tasks to achieve the compliance of RAMS for ongoing monitoring and review.

All these systems lead to one important concept of "Risk". Risk is defined in two different ways

1. "Risk is the combination of two elements of the expected frequency of occurrence of consequence (loss) of a hazard and the degree (severity) of the consequence." (BS EN50126-1, 1999).
2. "Risk is the likelihood that a hazard will cause its adverse effects, together with a measure of the effects." (Chen, 2012).

For successful Risk Assessments, the authority must collect relevant data from different sources; past failures, engineering models, expert judgment or experimental data. Many different Risk Assessment techniques have been developed over the years, however, the process itself poses many different problems and challenges to successfully implement the Risk Assessment e.g., at a Level Crossing. Some common problems associated with Risk Assessments are

1. The collection of data is difficult and in most cases is inaccurate.
2. Often the quantitative risk assessment is required, however, it requires very accurate data for its successful implementation. It is expensive to gather accurate numerical data and ensure the data is accurate.
3. The qualitative assessment relies on judgment, opinions and estimations; hence, it is subjective and not so reliable for Risk Assessment when compared with quantitative Risk Assessment.

2.4.2 General Risk Assessment

The simplest method of Risk Assessment is divided into two main groups; General and Specific. The General category is assessed by the circumstances, data and resources. It is further divided into three categories; quantitative, qualitative and semi-quantitative. The Specific is analyzed for its design process and is further divided into two categories; top-down and bottom-up assessment. In the design phase (such as the proposed work for a Level Crossing application), qualitative methods are often used as a preliminary risk assessment to obtain a general level of risk and estimate all possible hazards. The qualitative method is suitable for almost all possible failures and suggests safety monitoring for the design phase system. Often linguistic terms are used to describe the effects and frequency of the risks. The failure severity is often described as 'catastrophic, critical, marginal and negligible' and its frequency is mentioned as 'frequent, probable, occasional, remote and improbable'. The semi-quantitative measure is similar to the qualitative method and is used if the full quantitative measures are not available. The quantitative measure provides quantified measures to determine the alternative of the design of the system (An, Lin and Stirling, 2006).

In general, the Risk Assessment is defined qualitatively and quantitatively by defining these four aspects of Risk; Cause, Hazard, Consequence and Probability. The Risk Assessment generally address some fundamental questions as mentioned below

1. What can happen and why (by identifying risk)?
2. What are the failure effects (by defining the severity of the consequence)?
3. How likely is it to happen (by defining the frequency of occurrence of failure)?

4. What is the level of risk?
5. Is the risk tolerable or acceptable and is any further control required (by applying risk assessment techniques)?

2.4.3 Specific Risk Assessment

The Specific Risk Assessment is applied to analyse the consequence of failure effects. The assessments of such failures are achieved by identifying the root cause from past failure data and continued risk assessment to avoid or lower the risk associated with the system hierarchy. The system continues to assess the failures unless every single fault or failure causes and modes are identified (An, Baker and Zeng., 2005).

Different techniques available for Risk Assessment are preferred according to their applicability in different stages of the design. For example, a technique may not be available for Risk identification but is preferred for Risk Evaluation and other techniques are preferred for Evaluation but not for Identification. For this work, the technique called FMECA is preferred which provides flexibility in its applicability for all stages of Risk Assessment; Identification, Analysis and Evaluation.

Failure Mode Effects and Criticality Analysis- FMECA identifies, prioritize and control all failure modes that may include in the system design and process. It also adds the frequency of occurrence for each severity along with its severity level as mentioned in the standard MIL-STD-1629A-1980. FMECA is a systematic approach for the system design process especially if the system is complex. It is used to ensure all failure modes of the system are referred and enlisted to provide sufficient information for RAMS management. It provides details about the plan, function and maintenance of the system (Ebeling, 2004). John Moubray in his work (Moubray, 2001) suggested six essential questions for effective FMECA

1. How can a system fail?
2. What is the mechanism for the failure modes which occur?
3. What is the consequence of a failure mode?
4. What are the effects of the safety of a failure mode?
5. How can the failure mode be detected?
6. How can the design for a failure mode be supported?

In summary, the FMECA techniques are used to analyse the design phase, process phase and system phase. The design phase is used to analyse and consider any failures during the designing of the system, whereas the process phase considers the failure points during the manufacturing, processing or maintaining phase. The system phase considers the whole system and considers any failure points within the system. The mentioned questions are addressed either top-down or bottom-up approach, where top-down is a functional analysis of the whole system and bottom-down (often referred to as hardware approach) is used to make decisions in the design phase.

For the proposed work, the qualitative assessment for general and FMECA techniques for specific Risk Assessment is considered sufficient to provide enough information for RAMS management. For more details and surveys on other techniques please refer to the work in (Park, 2014).

3 DATA & ANALYTICAL ALGORITHMS

Two most important aspect of the present work is the sensor and the algorithm itself, therefore, the present section will discuss them in detail before discussing the Dataset specific to CCTV and RADAR.

3.1 OBSTACLE DETECTION SYSTEM

The Obstacle Detection System has some pre-requisite that should be in accordance with the standards set by Network Rail or other relevant organisations within the Railway Industry. These particular standards or requirements define the detectable area or the smallest object that should be detected at a Level Crossing site.

Area of Detection is defined as the "part of Level Crossing" that should be scanned for any obstacle present during its operational cycle. The total area of detection by Obstacle Detector installed at a Level Crossing is called *Crossing Area*, hence the name "Level Crossing Area". The area that should be scanned by Obstacle Detector is further divided into two main categories; Train Protection and People Protection. The Train Protection area should look for any obstacles present at and around a Level Crossing, like obstacles hanging on barriers and small objects causing derailments e.g., steel sheet etc. The area for People Protection should detect pedestrians either standing or fallen on a Level Crossing within the defined area.

For the pre-defined area of Train Category, the system should detect vehicles with a minimum size of 1500mm * 500mm * 1000mm or larger and ignore any small objects with a size of 250mm * 250mm * 250mm. The speed of such detectable objects should be a minimum of 4ms. Currently, no traditional sensors can detect smaller objects than 0.5m tall, hence, no pre-defined size is available for objects that may cause a derailment. However, in theory, it should not be smaller than 0.5m and a minimum of SIL 3 category.

For the People Category, the system should detect pedestrians of Minimum Detectable Object Size Standing Person (MDOSSP). An approximation of such object is 1245mm * 245mm * 150mm and shall ignore any smaller objects of size 100mm * 100mm * 100mm of speed less than 4ms.

An Obstacle Detection System should detect multiple areas of detection at a Level Crossing pre-defined area for any objects present. The Obstacle Detection System should have the ability to tailor its tools and specification for each Level Crossing type without installing new systems. The System should scan the entire "Area of Crossing" with the least number of sensors. A typical Level Crossing is about 11m * 15m, where a larger Level Crossing area is about 30m*18m.

The SIL rating, which is a numerical approach for the Safety Critical functions for Train and People Protection and it should be certified in accordance with EN 50128 and EN 50 129 respectively, where Train Protection should have a minimum of SIL 3 and People Protection of SIL 1. Every number of SIL represents the severity and frequency of these failures for its specific application e.g., Level Crossing. The individual sensing system and its proposed software e.g., Deep learning may have a smaller SIL rating of 1 or 2, the final output given by the Interlocking System will have a SIL rating of about 3 or 4. Also, the interfaces connections should be under NR/L2/SIG/30027 standard as well. The supplier should demonstrate that the design minimises the risk of injury to members of the public as far as is reasonably practicable under the Health and Safety Act (Jones, Vine and McManus, 2018).

3.1.1 Sensors

Earlier, a detailed discussion about sensors and their associated algorithms gives the reader an understanding of the functionality and adaptability of these sensors for their application at a Level Crossing. The proposed and preferred sensors should comply with the requirements of the Obstacle Detection System as discussed earlier. The traditional sensors have some limitations when used as a single sensor for its application at a Level Crossing because they cannot provide sufficient information for effective detection and classification of an object present at the site. However, if another secondary system is installed it shall add another layer of resilience and increase safety and providing a "2oo2" Interlocking system as well. Also, the associated algorithm is not reliable for its application at a Level Crossing since they rely on the "subtraction from the background pixel" method, which is an inefficient approach for a dynamic environment of a Level Crossing. Hence, the sensor proposed should be integrated with a new effective method or algorithm which can detect and classify objects based on their features rather than pixels. The sensor itself should be cost-effective in its installation and maintenance. The sensor should provide sufficient information for the proposed solution to learn enough features for classification and detection without relying on traditional methods. These sensors should also provide information for Risk Analysis, an essential key feature for industries to process and analyse information for the sustainability of their products.

The supplier of such sensors should provide details of their specific sensors e.g., the height, position, number of sensors required, any non-detectable area, range, the field of view or measurement accuracy of such systems. A perfect sensing system should have these features

1. The system should detect obstacles irrespective of the Level Crossings' furniture colour, density, shape, temperature and reflectivity.
2. The system should not be affected by environmental conditions such as rain of 15mm/min and hailstones up to 15mm.
3. The system should not be affected by other weather conditions like fog, mist, direct sunlight.
4. The response time from failure to no-failure should be about 1 second.
5. Each installed sensor should have its installation, position, safety and design in accordance with their respective standards.
6. The system should have internal monitoring features accessible over a Diagnostic Data Communications connection as well.

The proposed system should have these features or alternative methods should be used to overcome any limitations (if present). The service life of the Obstacle Detection System should be about 35 years, where if needed, maintenance services are provided by the supplier. The supplier should comply with NR/L2/RSE/0005 for the system's reliability. Also, the supplier should consider and provide adequate information about system availability, maintainability and any other physical requirement e.g., required space, insulation or bonding etc.

To discuss the proposed solution, it is essential to mention the hardware and the operation for the proposed sensors e.g., CCTV and RADAR. The choice of these two sensors is evident for two obvious reasons; they are already implemented at most of the Level Crossing sites within Great Britain and secondly, the low implementation cost, low maintenance and high product lifecycle make these two sensors the primary choice for a Level Crossing Application. Once the hardware and operation of such sensors are mentioned, the Deep Learning technique will be discussed in detail to further elaborate on how it works and why it is an effective solution for its application at a Level Crossing.

3.1.1.1 CCTV

The working principle of CCTV installed at a Level Crossing is fairly simple. The video feed is constantly fed to the monitor and supervised by a manual operator. The manual operator is responsible for the decision whether the Area of Detection is cleared for automation of the operational cycle or alarm for a high-risk situation if the object is present at the site. To ensure the system is working properly, there are a series of tests that are correlated with the competency levels identified in NR/L2/SIG/30014/B410. These test specifications are as follows

3.1.1.1.1 Verification

The make, model and serial number of the camera housing, lens assembly and power supply are recorded and inspected for any physical damage. Earthing and lightning protection is tested and inspected according to the given diagram and specifications from the company.

3.1.1.1.2 Hardware Test

The testing and installation are checked by competent Testers for Location wirings and platform tests. Afterwards, the video system is checked by applying power to it and connecting the camera with a monitor to ensure the video output is working as well. Before mounting and closing the housing unit, the video system is checked for Focus by dismounting and readjusting the focus on distant objects if needed. The heating test is also carried out before the final installation and mounting of the camera.

3.1.1.1.3 Function Test

The Camera alignment is an essential part of the Function test, where the camera is mounted in such a position and direction that it should cover the whole Area of Detection. The view should be visible and does not smear with moving objects. The Video Relay and Transmission Unit is tested according to the given specifications. Specified CCTV transmission medium is installed following relevant installation specifications. This should ensure the transmission system is working properly. The monitor is tested according to the manufacturer's manuals. The power supply, brightness, contrast, controls and lightning are tested before their installation as well.

Also, some other tests are carried out e.g., the lighting conditions and illumination levels at a Level Crossing area (Network Rail, 2012; Spowart, 2014), which ensure that sufficient light is provided for its operation during low light and adverse weather conditions.

3.1.1.2 RADAR

The RADAR is used as a primary sensor for Obstacle Detection systems. The RADAR uses a frequency of 24GHz, which is also the frequency of radio astronomy hence additional screening is necessary. The screening should be carried out if the RADAR lies within 20km range of other Radio telescopes (currently 6 in the UK). The RADAR can scan an area of over 110 degrees; therefore, the radio telescope is cleared of 110 degrees sector scanned by Obstacle Detector or should not be in clear sight to it.

Traditionally, the container with RADAR is placed at one corner of a Level Crossing and is placed in such a manner that 110 degrees can cover an entire Area of Detection. The other three corners are with RADAR reflectors. These reflectors act as a reference point for their operation. The RADAR sends a signal and looks for these reference points if any of these three reference points (RADAR reflectors) are missing or moved or hidden, the RADAR would not clear the Area of Detection and signal it as “occupied”. The detector is given a “start” command, which initialises the closing operational cycle. The system requires 8 seconds to warm up and then starts scanning at a rate of 1 scan per second. If three consecutive scans are cleared, the RADAR sends a “clear” signal for the barrier to be lowered. Once the barriers are in the lowered position, there is a short delay to allow the sensor for one final scan before sending the sensor a “stop” command. The “stop” command allows the system to stop the scanning process and complete the normal operational cycle (Network Rail, 2012).

The proposed method does not rely on the traditional method of RADAR working principle, rather its post-process the signals from RADAR. Post-processing is required to obtain micro-Doppler signals which act as a pixel image for the Deep Learning algorithm. The Convolutional Neural Network (as discussed later) will learn representations from these micro-Doppler signals for classification. The post-processing is computed using the existing installed system at each Level Crossing, which means no additional cost is incurred.

3.1.2 Algorithm

3.1.2.1 A Brief History of Neural Network and Convolutional Neural Network

A brief history of Neural Network is presented followed by a detailed discussion on Neural Network and Convolutional Neural Network before discussing and visualising the results achieved by Convolutional Neural Network for its application at a Level Crossing.

Machine Learning powers many aspects of modern society; from web searches to recommendations on e-commerce websites and it is increasingly used in consumer products such as smartphones and cameras. Neural Network is a Machine Learning algorithm built in the 1970s to extract information from the data. The most common type of Neural Network in use was Feedforward Neural Network, which processes information only in the “forward” direction (hence the name) to extract information and map output from the given input. This simplest Feedforward Neural Network is connected via “nodes” with only Single Layer Perceptron (Schiopu, 2009), where output is mapped directly from an input. However, the Single Layer Perceptron can only be used for linear problems and could not learn non-linear complex problems like classification or pattern recognition.

Since it was unable to learn complex features, the idea of learning from Neural Network was not well celebrated (Minsky and Seymour, 2017) until 1986, where Geoffrey Hinton (often called Godfather of AI) published their work (Rumelhart, Hinton and Williams, 1986), which discuss the idea of “backpropagation”. The “backpropagation” demonstrated that Neural Networks with multiple layers can learn a complex function from a relatively simple procedure. The “backpropagation” contributed to the work in (LeCun *et al.*, 1990) introduced a Convolutional Neural Network called LeNet-5. The work demonstrates how well the “backpropagation” works in automating the recognition of handwritten characters but it also introduced few essential layers of Convolutional Neural Network architecture; Convolution and Pooling Layer (as discussed later). The backpropagation method in Machine Learning was forsaken in the 1990s because it was thought that learning useful, multistage, feature extractors with little prior knowledge were not feasible. In particular, it was thought that the gradient descent would be trapped in local *minima* (weight configuration, where no small change in the backpropagation algorithm would reduce the average error) called *vanishing gradient*. In practice, it was hardly a problem with large networks and regardless of initial conditions, the network nearly always reaches solutions of similar quality. Therefore, the Support Vector Machine- SVM becomes a preferable choice for computer vision tasks (Géron, 2019).

Later, Geoffrey Hinton proposed another work called Belief Nets (Hinton, Osindero and Teh, 2006), where the work demonstrated how stacking layers of perceptron can learn more abstract and complex features. This stacking of layers gives an idea of "*deep*" learning, hence the name and field of Deep Learning emerged (LeCun, Bengio and Hinton, 2015). The interest in Deep Learning grew exponentially when many novel ideas and results were achieved in different applications such as Speech Recognition (Hinton *et al.*, 2012) and Object Classification (Krizhevsky and Hinton, 2012). Geoffrey Hinton again managed to submit a winning model in the well-known ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with his research fellows. The ILSVC classify objects using a dataset from ImageNet, which is used for this work as well in its application at a Level Crossing (Russakovsky *et al.*, 2015). The winning model created a transition in research from *feature engineering* to *Deep Learning*.

Deep Learning has been widely used for three main important reasons (Deng, 2014)

1. Availability of large-scale annotated training data such as ImageNet, which means that achieving the full learning capacity of the Deep Learning network was possible.
2. Progress in chip processing power such as high-performance parallel computing (GPUs)
3. Significant advances in the design of Neural Networks and training strategies. For example, a good initialization is provided with unsupervised and layer-wise pretraining. The use of dropout and data augmentation techniques to overcome the overfitting problem or training Deep Neural Network using batch normalization techniques is more efficient (Hinton, Osindero and Teh, 2006; Hinton *et al.*, 2012).

Within the Deep Learning field, the researchers significantly benefited from the use of GPUs to compute the backpropagation more quickly and efficiently to achieve more accurate results. The use of GPUS also meant deeper networks were possible. The GPU made many new developments within the field of Deep Learning possible e.g., Dropout (Srivastava *et al.*, 2014), and ReLU (Nair and Hinton, 2010) (used to solve vanishing gradient problem). All these developments ensure the continuous growth of Deep Learning and ground-breaking research in different applications.

Many applications within Deep Learning use a feedforward Neural Network, where a fixed input (image) is the map to a fixed size output (set of categories). From the first layer to the second, the Neural Network computes the weighted sum of the inputs from the output of the previous layers and apply a non-linear function. The most commonly used non-linear function is Rectified Linear Unit (ReLU), which learns faster in networks compared with other non-linear functions like tanh(z) (Glorot, Bordes and Bengio, 2011). The first layer is called the input layer, the last layer is an output layer and in between layers are called hidden layers. These hidden layers are seen as distorting the input in a non-linear way so the categories become linearly separable by the last output layer.

Deep Learning techniques are further divided into four main categories; Convolutional Neural Network-CNN, Restricted Boltzmann machines (RBMs), Autoencoder and Sparse Coding. One particular type of feedforward Neural Network which was able to generalize a lot quicker and train easier compared with other fully connected networks is Convolutional Neural Network (CNN) (Lecun *et al.*, 1998).

Since the early 2000s, CNN has been applied with great success to the detection, segmentation and recognition of objects and regions in an image. These all tasks were possible with the availability of labelled data in abundance, such as traffic sign recognition (Cires and Meier, 2012), detection of faces, pedestrian and human bodies in natural images and face recognition (Vaillant, Monrocq and Cun, 1994; Garcia, Delakis and Intelligence, 2004; Osadchy, 2007; Turaga, Murray and Seung, 2010; Tompson *et al.*, 2015). Despite these successes, CNN was largely forsaken by many computer vision practitioners until the ImageNet competition in 2012, which was trained on millions of images and achieved spectacular results (Krizhevsky and Hinton, 2012). These signs of progress and success came from the use of GPUs, ReLUs, a new optimization technique called dropout (Srivastava *et al.*, 2014), and techniques to generate more training examples by deforming the existing ones. Some of the advantages of CNN over traditional Neural Networks are as follows

1. Multilevel representation from pixel values to high-level semantic features are learned automatically from multistage structure through multilevel nonlinear mappings.
2. Deep Networks provides more expressive capability compared with shallow networks.
3. The architecture of CNN provides an opportunity to optimize several related tasks together. For example, the Fast RCNN is a combination of classification and bounding box regression methods (Guo *et al.*, 2016).

There have been some developments in the CNN algorithms particularly in the Computer Vision field, where some well-known models have emerged. AlexNet (Krizhevsky and Hinton, 2012) model won the ILSVRC2012 competition and set a tone for the surge of interest in the CNN domain (Russakovsky *et al.*, 2015). The AlexNet model consists of 5 Convolution layers and 3 Fully Connected layers (these layers are discussed later). The network's input is a fixed resolution image of size 224*224, where the image is passed through a Convolution and Pooling layer before being fed to the Fully Connected layer. The problem with such a network is Fixed Resolution and No clear understanding of why it performs so well. To deal with the Visual Understanding of how the network performs, the work in (Zeiler and Fergus, 2014) introduced novel Visual techniques to get an insight into the network. To overcome the fixed resolution problem, the work in (He *et al.*, 2014) introduces a new pooling strategy called Spatial Pyramid Pooling. Despite the commonly used CNN algorithms, different researchers introduced different models for different applications. For example, the VGG model (Simonyan and Zisserman, 2015) increases its depth and uses small Convolution filters. The GoogleNet (Szegedy *et al.*, 2015) also introduced a very deep Neural network (22 Layers), which achieved leading performance in the ILSVR2014 competition.

For applications other than classification such as Object Detection and Segmentation different frameworks have been introduced. The most commonly used framework for Object Detection is RCNN (Girshick *et al.*, 2014) and for Image-Segmentation is FCN (Long, Shelhamer and Darrell, 2015). The RCNN model generates multiple object proposals, which is used to extract features from each proposal and classify each window using Support Vector Machine (SVM). The problem with such algorithms is that it is inefficient and also the object must lie within the proposed region, which reduce the robustness of the model. FCN models used the recasting techniques, which remove the restriction on image resolution and efficiently produce corresponding-size outputs. Even though, the FCN models are trained for Image-Segmentation it could be used for other applications like Image Classification and Edge Detection.

There are other characteristics of the CNN architecture such as Large Networks or Multiple Networks (Sun, Wang and Tang, 2013; Wang *et al.*, 2014; Ouyang *et al.*, 2015). The idea behind the Large Network is to make the CNN algorithms deep such as GoogleNet with 22 layers or add a network for different tasks in cascade mode. For example, the work in (Wang *et al.*, 2014) proposed a two network model for Object Localisation, where the first network is used for Object Localisation and the second network is used to output coordinates of an object. Similarly, the Multiple Network combines the result from multiple networks and combine the results to give an estimation. For example, the work in (Cires and Meier, 2012) proposed a method called Multi-Column DNN (MCDNN), which combines several DNN columns and average their output predictions to give an estimate. The model achieved an accuracy of human-level estimation on hand-written digits.

From the discussion above, the importance of the Machine Learning field is obvious in particular the sub-field of Machine Learning called Deep Learning. Within the field of Deep Learning, the Convolutional Neural Network is the leading architecture to train models for Object Classification, Detection and Segmentation. It is possible to integrate the Deep Learning techniques with the existing sensing system at a Level Crossing site e.g., CCTV and RADAR. The ability to adapt the data structures from CCTV video cameras and RADAR signals makes the proposed solution more unique and efficient. The backbone of each model for classification and detection is a CNN architecture, so it is essential to discuss Neural Networks and Convolutional Neural Networks in detail before discussing its application at a Level Crossing.

3.1.2.1.1 Neural Network

The Neural Network is the foundation of Deep Learning models and the building blocks of each Neural Network are its “neurons” or “nodes”. The idea of such nodes originated from the “Neurons from Human Brain”, where each neuron is responsible for receiving and sending a signal to multiple other neurons for transmission of information or stimulation of a response. The working principle of a node is based on an artificial neuron called the Linear Threshold Unit (LTU) (Rosenblatt, 1957), where the input and outputs are numbers. The output is connected with all multiple inputs with some associated value called weights (w). The weight parameter defines how strong the influence of that particular input is compared with other inputs. The final output of the node is the weighted sum of all the given inputs as shown in Equation 1.

$$z = w_1 x_1 + w_1 x_1 + \cdots + w_y x_y \qquad \text{Equation 1}$$

For example, the output required from such a network is whether the train should pass the Level Crossing or stop in case of high-risk situations. The input for such networks could be “Is the Level Crossing area empty” or “time of day”. Now the input “Is the Level Crossing area empty” should have more influence compared with “time of day” or maybe the “time of the day” is irrelevant and the value assigned to such input might be 0. Once trained, the network will give preference to the first output rather than the second to make the right decision. These inputs are called *features* of the Neural Network, which according to their relevance are assigned *weights* to give certain output. A function is applied to the given output to give results as shown in Figure 13 (Picture Courtesy: (Gaudet, 2016)).

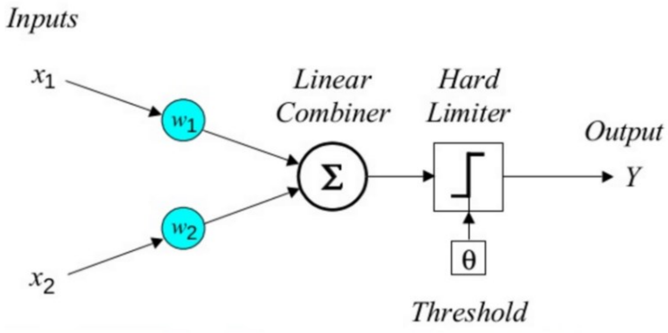


Figure 13A visual representation of LTU, where inputs according to their weights give certain output. The output is further combined with a function (Hard Limiter in this case) to give the final output.

Each node is connected to every single node from the previous layer, such connection of nodes is referred to as "*Fully Connected Network*" (each output is fully connected with all inputs). Such a single layer Network cannot perform an exclusive XOR operator (Minsky and Seymour, 2017) used in multiple classification problems. And since the network was unable to learn such logical functions, the research within Neural Network stalled but regained its fame when the idea of "*Multiple Perceptron Layer"* was introduced. The Multi-Layer network can learn and perform logical functions such as XOR used for classification problems. The network consists of multiple fully connected layers except for the final output layer. Each layer has nodes connected with previous nodes of the previous layer with some value of *weights* and *bias* (bias is another parameter added to shift the output value for a better fit with input value) (Rosenblatt, 1958; Ruck *et al.*, 1990). Such a network can learn complex functions required for classification problems, and the depth of these networks gave rise to the "*Deep" Neural Network*. A simple Multi-Layer Neural Network is shown in Figure 14 (Picture Courtesy:(Nielsen, 2019b)). The "Hidden Layers" are not directly accessible in the network, hence the term "hidden".

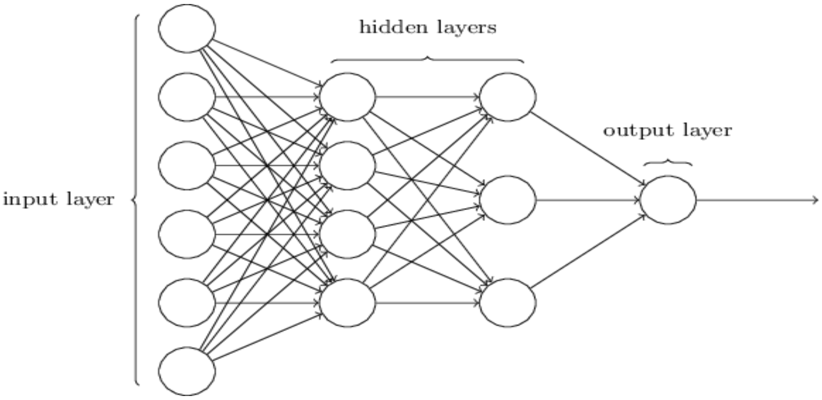


Figure 14A simple Multi-Layer Neural Network. The network has 6 inputs and 1 output layer with 2 hidden layers.

3.1.2.1.2 Backpropagation

The introduction of the "backpropagation" method was introduced by Hinton in (Rumelhart, Hinton and Williams, 1986), which allowed the networks to learn even more complex representations from multiple layers. This was essential for complex applications like classification, detection or pattern recognition. The main purpose of backpropagation is to find the *gradient* of an error with respect to its *weights*. The *gradient* determines the value by which the error changes with respect to the change in the *weights* of the network. The ultimate goal is to minimise the error of the network by optimizing the value of the *weights*. The authors of the work (Rumelhart, Hinton and Williams, 1986)changed a key feature of the function applied before giving a final output called an activation function. The traditional Multi-Layer Network used a step function, whereas the new proposed function is the continuous activation function. The need for activation function is obvious once the working principle of backpropagation is clear. The backpropagation finds gradients to minimise the error by optimizing the value of weights, which is achieved from *gradient descent*. The step function has constant gradients; hence it is impossible to find gradient descent that changes and optimizes the parameters using backpropagation method. However, the activation function is a continuous value which makes it possible to find gradient descent, allowing the network to change values at each step.

The backpropagation algorithm finds an absolute minimum of a loss function. The real-world problem is not a simple 2D curve graph, where the minimum is one and easy to calculate rather the real-world has multiple minima and a more visual representation would be a 3D curved surface as shown in Figure 15 (Picture Courtesy: (Amir, 2015)).

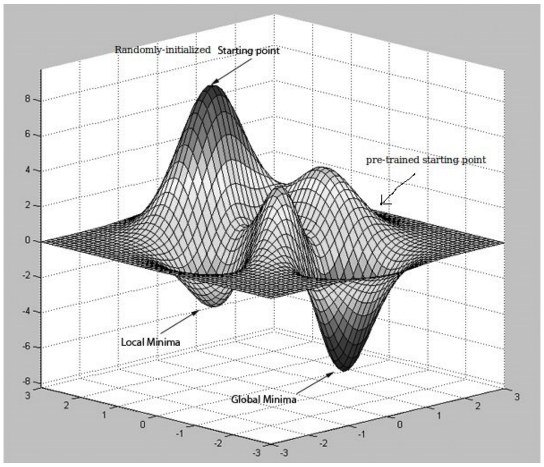


Figure 15A visual representation of how backpropagation algorithm tries to find a global minima. The initial point is randomly initialized and updated accordingly in the steepest route.

This suggests that the model may change in such a way that instead of finding a minima point, it ends up further away from the minimum in the opposite direction. To avoid this situation, the backpropagation method should find the steepest route to the minima, reach the minima point, reassess the situation and find another (if any) steepest point for minima until it reaches global minima. Traditionally, the error gradient was calculated on the whole training dataset at once which was a computationally expensive and time-consuming process. To avoid this situation, the use of *Stochastic Gradient Descent-SGD* algorithm is used. The SGD allows the Neural Network to take a subset from the data, make a decent estimation of the error and calculate the gradient descent to update the values in the global minima direction. The size of each step in this process is determined by another parameter called *Learning Rate*, which is often smaller than used in the traditional approaches. Another common problem with such an approach is the presence of many "*Saddle Points"*, which are flat dimensions where the value does not change much but a deep route is available nearby. This suggests that when a gradient descent reaches a flat surface, the update values are small and the progress is very slow. Therefore, it is easy to find a good solution for a problem (since there are multiple minima) but it is hard and time consuming to find the best solution (the global minima). For a mathematical explanation of how backpropagation works please refer to the work of Michael Nielson (Nielsen, 2019a).

3.1.2.1.3 Activation Function

The output for our Neural Network is given in probability, therefore, it is common to practise to apply some function that scales the value between 0 and 1. This suggests that the output will be very non-linear compared with simple linear problems. The nonlinearity comes from non-linear preferences of input weights by the Neural Network during the training process (Goodfellow, Bengio and Courville, 2016). For more complex problems e.g., classification and pattern recognition, the hidden layer's output is non-linear as well. Some most commonly used Activation functions are briefly discussed here.

3.1.2.1.3.1 Sigmoid Function

The most simple and traditional Neural Network was used for binary classification, where the output is simply "yes" & "no" or as stored in binary outputs "0" or "1". This means that a Neural Network can use a simple logistic sigmoid function, which uses a threshold value to classify the output as either 1 or 0. For example, a simple Neural Network consists of some inputs as features, a single output and a threshold value of 0.6. If the features determine the value of output to be greater than the threshold value (e.g., 0.6) the output will be 1 otherwise 0. The sigmoid function is represented by following Equation 2

$$f(z) = \frac{1}{1 + e^{-z}}$$ Equation 2

3.1.2.1.3.2 Hyperbolic Tangent Function

The hyperbolic tangent function works on the same working principle as the sigmoid function except it has a different range between -1 and 1. Such functions are used in regression problems, where an image has pixel values ranging within the given range of -1 and 1. The function is often used in hidden layers to add non-linearity (Glorot, Bordes and Bengio, 2011) and the function is defined in the following Equation 3

$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$ Equation 3

3.1.2.1.3.3 Softmax Function

Often the problem does not require a single output but multiple outputs e.g., multi-class classification problem. In the multi-class classification problem, the Network is trained on multiple classes and requires an output that strongly correlates to one particular class from these multiple classes. The Softmax function will apply an exponential function to each output, then it will divide the output by the number of classes (represented by value K) to normalize the output. This would mean that the Softmax function not only gives an output in a range from 0 to 1 but also the sum of all given outputs is equal to 1 as well. Therefore, the Softmax function is used where a vector is required to represent the multiple outputs for each given class, whereas the sigmoid function is used to give one single output for binary classification. The Softmax function is given in the following Equation 4

$$f(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \qquad \text{Equation 4}$$

3.1.2.1.3.4 Rectified Linear Unit

In an Image Classification problem, the Neural Network depends on the features it learns during its training process. The neurons in the output layer are connected with every single input neuron of the previous layer. This means often a feature is learned in one particular neuron, which might not be useful for the next layer but it is still transferred costing time and computation. To solve such an issue, the Neural Network use another activation function called Rectified Linear Unit, which essentially convert all negative values to 0. The function is defined in the following Equation 5

$$f(z) = \begin{cases} z & when\, z > 0 \\ 0 & when\, z \leq 0 \end{cases} \qquad \text{Equation 5}$$

This simple function will convert all negative values to 0, hence, during the backpropagation, any negative values (which are 0 now) will not be used to update the weights. Also, the positive values will give higher gradient values compared with any other activation function. This means that the ReLU function will achieve better training and computing speed compared with other activation functions. For these reasons, it is a common practice to use ReLU as a default activation function for Neural Networks (Noh, Hong and Han, 2015; Cao *et al.*, 2017). Recent studies like in (Glorot, Bordes and Bengio, 2011), shows that the ReLU performs better for hidden layers compared with other activation functions like sigmoid and tangent functions. Another improved version of ReLU is proposed in (Xu *et al.*, 2015) called leaky ReLU. Leaky ReLU essentially adds a “small value” to the output to avoid any information loss by controlling the number of negative values that could pass to the next layer. Another work in (He *et al.*, 2015) proposed the use of parametric ReLU, which makes the “small value” added in Leaky ReLU a trainable parameter.

(For more details on activation functions and their derivatives please refer to the work in (Géron, 2019))

3.1.2.1.4 Loss Function

As mentioned earlier, the Neural Network uses backpropagation to optimize the parameters such as weights to minimize the error between the ground truth and predicted value. The “error” also known as “Cost Function” determines how accurate the Neural Network is at that particular moment in the training process. Accordingly, the cost function uses gradient descent to update the weight parameters by taking a partial derivative of the Loss value with respect to its parameters. This allows the Network to update its *weights*, which in turn will minimize the loss between the predicted and ground truth value. Different Loss functions are used for different applications and preferred accordingly.

3.1.2.1.4.1 Logistic Regression Loss

A simple binary classification problem has only one output with binary values (either 0 or 1) from a simple sigmoid function as discussed earlier. The Logistic Regression Loss function should guide the network towards a single output of either 0 or 1 based on their feature values (inputs). The loss function is defined in Equation 6, where y is a binary output and y represent the predicted value and “m” are the number of classes. Logistic Regression Loss function minimizes the value of y and $\hat{y}$.

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}(y^i \log(\hat{y}^i) + (1-y^i))\log(1-\hat{y}^i) \qquad \text{Equation 6}$$

3.1.2.1.4.2 Mean Squared Error

Another simple Loss function is Mean Squared Error-MSE. The MSE is used for regression problems where the output is not one binary value rather a continuous value like pixel values of an image or a scalar value. The function is defined as an averaged $\mathcal{L}_2$ norm and is represented by Equation 7

$$\mathcal{L} = \frac{1}{m}\sum_{m}^{i=1}(y^i - \hat{y}^i)^2 = \frac{1}{m}\|y-\hat{y}\|_2^2 \qquad \text{Equation 7}$$

3.1.2.1.4.3 Mean Absolute Error

Similar to the MSE function, the Mean Absolute Error-MAE is a loss function and is defined as an L1 average norm. In principle, it is simply the difference between the output and the predicted value. The function of MAE is given in Equation 8

$$\mathcal{L} = \frac{1}{m}\sum_{m}^{i=1}|y^i - \hat{y}^i| = \frac{1}{m}\|y-\hat{y}\| \qquad \text{Equation 8}$$

The Mean Squared Error is a simple mathematical problem; hence it is easy to compute and does not require high computational power for partial derivation during the backpropagation method. However, the Mean Absolute Error is finding the “absolute” value which is computationally extensive. Another key difference between this two-error function is when the difference between y and $\hat{y}$ is 1. For the MSE function, the error will be very large compared with the MAE function predicting this particular value and is more sensitive for the network training process.

3.1.2.1.4.4 Cross-entropy Loss

Cross entropy is a widely used Loss function in Neural Networks. The Loss function avoids slow learning when an error is a large value, rather it learns fasters with greater loss values. Two different cross-entropy functions are available for Neural Networks; Binary and Categorical Cross-entropy. The Cross-entropy for binary and multi-class classification is given in Equation 9 & Equation 10respectively.

$$\mathcal{L} = -\,(\mathrm{ylog(p)} + (1-\mathrm{y})\mathrm{log}(1-\mathrm{p})) \qquad \text{Equation 9}$$

Equation 10

$$\mathcal{L} = \sum_{c=1}^{M} y_{o,c}\mathrm{log(po,c)}$$

The authors of Deep Learning (Goodfellow, Bengio and Courville, 2016) described the Cross-entropy as follows

"The use of cross-entropy losses greatly improved the performance of models with sigmoid and Softmax outputs, which had previously suffered from saturation and slow learning when using the mean squared error loss."

For more details on Loss functions please refer to the work in (Ml-cheatsheet.readthedocs.io, 2017; Truong, 2019).

3.1.2.1.5 Summary

The most common form of Machine Learning technique is Supervised Learning (used in Deep Learning models), where labelled data is fed to the network and it learns features to give an output in terms of a "vector score", one for each category. The purpose of training the network is to have the highest vector of a score for the desired output each time, and it is unlikely to happen before training. The network will be assigned an objective function that will measure the error or distance between the output vector score and desired output score. The function then modifies the parameters (often called the weights) to reduce the error. To reduce the weight error, the function computes a gradient vector that indicates by what amount the error would increase or decrease with a small change in weights. The weight vector is then changed in the opposite direction. The objective function averaged over all training examples, is like a hilly landscape in the high-dimensional space of weight values. The negative gradient (opposite direction of the gradient vector) indicates the deepest point in the landscape, where the difference between output score and predicted score will be minimal. Different optimization techniques are used to minimize the error but the simplest yet effective technique to optimize weights a lot quicker than many elaborate optimization techniques is "stochastic gradient descent SGD" (Bousquet and Bottou, 2007). Stochastic Gradient Descent (SGD) takes a small set of examples, computes the output and error, then computes the average of these examples and adjusts the weights. It repeats the procedure by taking another small set of examples until the objective function stops decreasing. The backpropagation procedure to compute the gradient of the weights for multiple layers is nothing more than a derivative using a chain rule method. The backpropagation equation is repeatedly applied to find the gradient of all modules starting from the top layer output to the input layer at the bottom, and once it is computed, the gradient for weights is straightforward (Dauphin *et al.*, 2014; Choromanska, Henaff and Mathieu, 2015; LeCun, Bengio and Hinton, 2015). The process is repeated until the Neural Network has learned sufficient information from the dataset and map the output from the given input quite accurately.

3.1.2.2 Convolutional Neural Network

The Convolutional Neural Network (CNN) is the most effective and commonly used algorithm for computer vision tasks, where multiple layers are trained to automatically learn features from an input image (Lecun *et al.*, 1998). A general pipeline of CNN architecture is shown below in Figure 16 (Picture Courtesy: (Rajan, 2017))

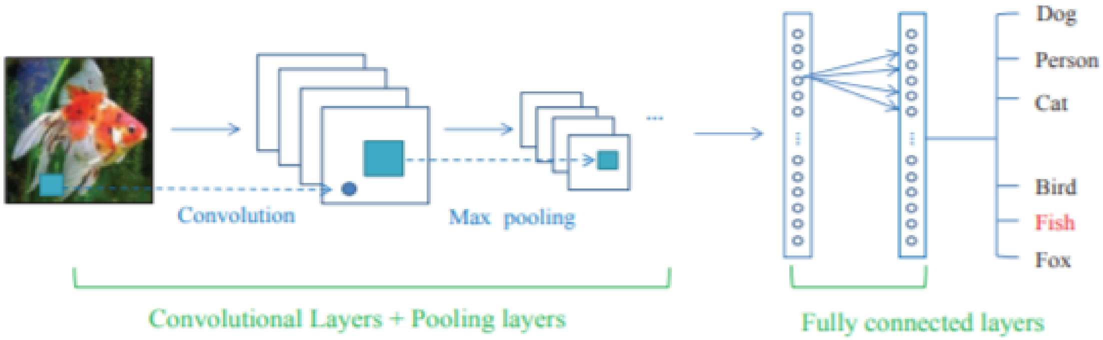


Figure 16A general pipeline of CNN architecture. It has an input layer, multiple hidden layers (typically a convolution layer followed by Pooling layer) and an output layer.

Traditionally, the CNN is composed of Convolution, Pooling and Fully Connected Layer (Brosch and Tam, 2015). It is hard to find the right topology network for your application but a general rule is to have a Convolution layer followed by a Pooling Layer and end the Neural Network with a Fully Connected layer. The filters used in these layers should be of small kernel value to allow the Neural Network to learn more complex features without losing too much information. The ability of Convolutional to be space invariant allows the number of trainable parameters to be significantly less than the normal Neural Network of the same size. This makes a deeper Neural Network possible, hence it is easy to train and learn complex features required for classification and detection (Krizhevsky and Hinton, 2012; Uijlings *et al.*, 2013; He *et al.*, 2016). The Convolutional Neural Network architecture is discussed below.

3.1.2.2.1 Convolutional Layer

A Convolutional layer uses a kernel to convolve the whole image or intermediate feature maps to output various feature maps used for predicting an object or new input layer for the next layer within the Neural Network (Zeiler and Fergus, 2014) as shown in Figure 17 (Picture Courtesy: (Hongchenzimo, 2018)).

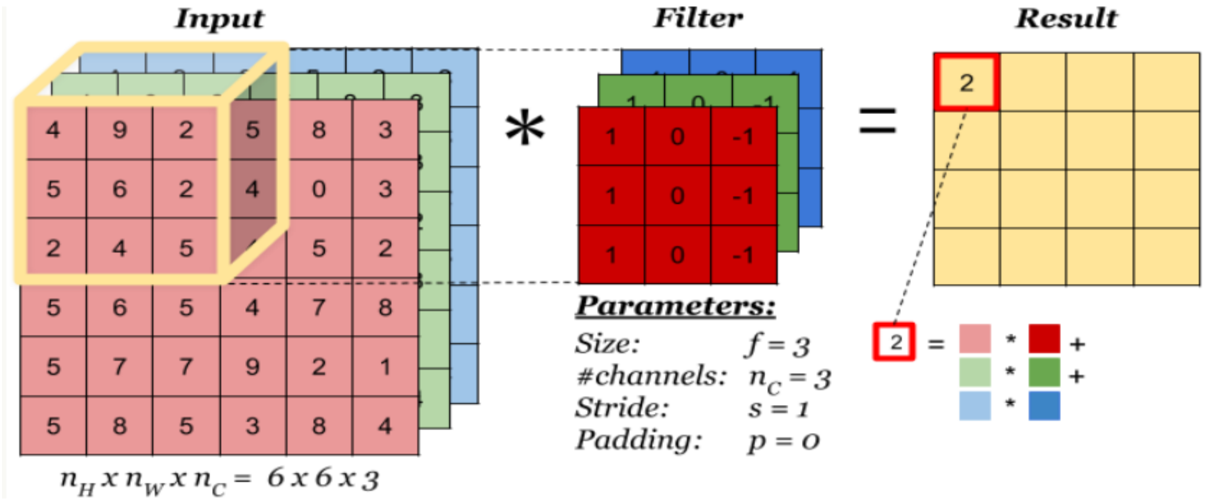


Figure 17A Convolution Layer, where a set of filters is performing matrix multiplication and adding the sum to given an output for "convolved image" or feature map.

The Convolutional Layer is the most important aspect of CNN architecture, where multiple convolution layers are stacked to learn more complex and abstract features required for classification. They can be visualised as “neurons'' similar to Neural Networks, but unlike traditional Neural Networks, they are only connected to a local rectangular spatial location of the previous layer. This allows learning of low-level features and connects them at a higher level in Neural Network architecture. Like traditional Neural Networks, the “neurons” are optimised during the training process by optimizing their associated value of weights and bias. In mathematical terms, the “kernel” filter using convolution operation is calculating how these two signals overlap as they pass over. For example, a simple edge detector computes a dot product and gives a summed output for that location. If the values of the edge detector and the image location match, the output will be a higher number indicating the presence of a “feature”. Similarly, the Convolution layer consists of many such detectors and each detector uses a “sliding window” to compute the dot product and summed output for each location within an image to indicate the presence of a particular “feature”.

These feature maps are used to recognise patterns or edges required for their classification at a Level Crossing. The name “Convolution Layer '' is from the operation it performs, which is to convolve a matrix using a kernel. A kernel will overlap the matrix (for example, an RGB image from a scene at a Level Crossing) and find the product of the numbers from the matrix and kernel at the same location, later the sum of the product number is calculated which is the output value of that particular position and layer in the CNN. The overlapping is continued until all possible positions are utilised. Suppose, the input for a particular CNN network is an image which is of order 3 tensor from a CCTV camera at a Level Crossing. To perform convolution the kernel or “filter” should be of the same size; 3 order tensors. The filter is overlapped on the input image. The products of numbers from the filter and an image at the same position is calculated. The sum of these product values is an output of that particular position in the “feature map”. Different “kernels/filters'' are used to find different patterns or features from an image. For example, the

Sobel kernels have been used to calculate the vertical and horizontal edges in an image, which is essential to find distinct edges of a car or pedestrian. The convolution operator gives a high output value for instances where these edges or features are present, making these particular features highlighted in the training process. Later some other filters might try to find patterns or edges in different angles or orientations. Further down the network, the model can activate and combine these features into more complex group features. These group features are further combined to make parts of a particular object which is ultimately used to classify an object (Dumoulin and Visin, 2016; Karpathy, 2018)

Another benefit of using a Convolutional Layer is "spatial sharing" and "parameter sharing". The "spatial sharing" allows the CNN network to classify and detect an object regardless of its position and orientation since the same filter is used in every position of an image which is essential for its application at a Level Crossing. The "parameter sharing" would allow the network to use the same filter to find low-level abstract features common to all objects e.g., edges, patterns. This parameter sharing would benefit the training process as the number of filters would be significantly reduced and would allow the same filters for multiple categories. Unlike the traditional approaches, the "spatial sharing" and "parameter sharing" features of the CNN network would allow the model to learn the features of an object and does not depend on raw pixel values. Also, the CNN would have a smaller number of trainable parameters and deeper CNN architecture are possible making it more effective for its application at a Level Crossing. This is often represented by "distributed representation", where a certain feature is used to classify all possible categories or all learned features are used to classify one particular object. Due to the benefits introduced by Convolutional Operator, the researchers are replacing the fully connected layer with the Convolutional Layer for the fast learning process (Oquab, 2012; Szegedy *et al.*, 2015).

The Convolution Layer requires pre-defining of some other parameters; Filter Numbers, Stride and Padding. These parameters are briefly discussed below

Stride

The use of a "sliding window" will allow the CNN network to utilise every single position within an image. However, the practitioner can change the "stride" value, which means the step it takes for the sliding window across the image. Value 1 suggests that it will move across every single position, whereas, value 2 suggest that it will skip one step and move to the second position to perform the convolution and so on.

Padding

As shown in Figure 17, the output is smaller than the input because of the Convolution operation. This could be avoided with "padding", which is essentially adding some values (the most common value used is "0") around the input image. This would ensure that the output image is of the same size as the actual input image. This is a necessary step if the user wants to learn more features using a deeper network, otherwise, the input image will quickly shrink and learning features from it would be difficult. Also, the input image with no padding will not allow filters to computer convolution operation at every single position especially at the edges.

Filter-Numbers

The number of Filters is defined for each layer of the Convolutional Neural Network. This defines the output volume since each filter would produce an output. The filters are used to extract features of a particular type from an image. These features are essential for learning more abstract and complex features necessary for classification.

For more explanation and mathematical representation of Convolution Layer, please refer to the work in (Skalski, 2019) or explanation by Andrew Ng in his course “Convolutional Neural Network”.

3.1.2.2.2 ReLU Layer

Often a ReLU layer is added next to the Convolutional Layer. The ReLU will increase the non-linearity in an image, which is important considering the real world around us is very non-linear. For example, an image of a person standing next to a vehicle at a Level Crossing is a very non-linear semantics of scene (Becker, 2018). The ReLU activation function is discussed earlier in the Rectified Linear Unit. The ReLU Layer does not have any parameters to learn or does not change the dimensionality of an input and an output of the layer. Graphically the ReLU function is mentioned in Figure 19.

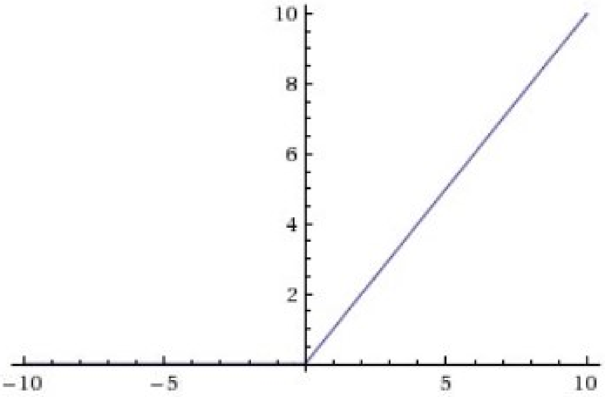


Figure 18Rectified Linear Function. The ReLU converts any given negative values to zeros.

Different non-linear functions are available for the Neural Network training and each is suitable for their particular application. For the present work, the ReLU function is used because of the details mentioned earlier.

3.1.2.2.3 Pooling Layer

The pooling layer is a straightforward layer with no trainable parameters to learn, hence backpropagation is straightforward as well. The pooling layer reduces the dimensions of an image, which will decrease the computation load and memory usage. Since it has no trainable parameters the risk of overfitting is reduced as well. It does not have any weight or bias but does require other parameters like *Stride, Padding and Filter Size*. The problem with traditional Neural Networks was their computational load and a high number of trainable parameters, these limitations are avoided from the use of the Pooling Layer.

The pooling layer reduces the dimensionality of the layer and since it takes the neighbouring pixels into account, the Pooling layer is also translation-invariant like Convolutional Layers. The model divides the input into subregions, where each sub-region gives one single value. The neuron in the Pooling Layer is connected to the output of a limited number of neurons from the previous layer, located within a small rectangular field. The Pooling Layer does not have any weights to train but a simple arithmetic operation like mean or average. The two most commonly used Pooling operators are; Max Pooling and Average Pooling. As the name suggests, Max. Pooling takes the highest value in the subregion, while the Average

Pooling takes the average value of the subregion. This Pooling Operation is shown in Figure 19 (Picture Courtesy: (Ricco, 2017)).

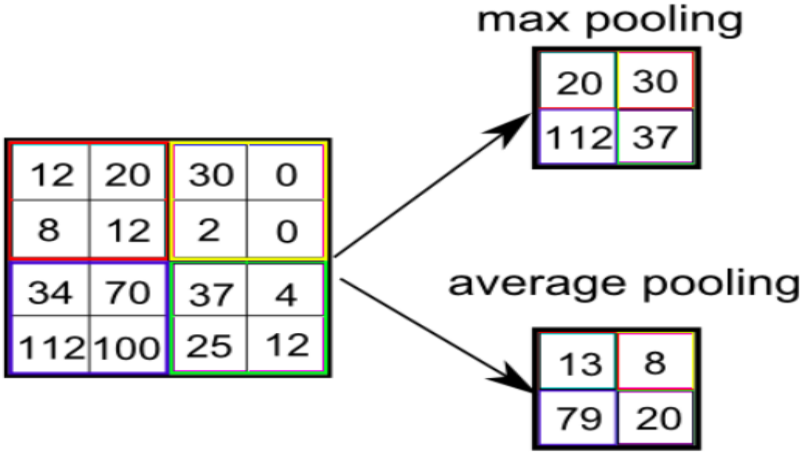


Figure 19Visual representation of how Pooling Layer works. The two most commonly used Pooling Layer is Max. & Average Pooling Layer.

Max. The pooling layer reaches convergence faster, selects superior features and improves generalisation as compared to Average Pooling (Boureau, Ponce and LeCun, 2009; Scherer, Andreas and Behnke, 2010). The Pooling Layer is the most studied topic among the three common layers of CNN architecture. Three well-known approaches for Pooling Layer are Stochastic Pooling (Zeiler and Fergus, 2014), Spatial Pyramid Pooling (He *et al.*, 2014) and Def Pooling (Ouyang *et al.*, 2015).

3.1.2.2.4 Fully Connected Layer

Once the features are learned by the previous layers of the Network e.g., from the Convolution and Pooling layer, the model can use all these learned features to make a prediction. A fully Connected Layer is used to accumulate all the distributed representation learned from the image and its layers, which is used to build features with stronger capabilities. The Fully Connected (FC) Layer acts as a traditional Neural Network, which contains about 90% of the parameters. The layer converts the 2D feature vectors into a 1D feature vector of a predefined length, which is used for the categorisation of the objects in Image Classification. Since the layer contains about 90% of the parameters the computational power required for such a layer is high. It is uncommon to change the Fully Connected Layer, however, for the transfer learning approach, it is common to practise to retain the parameters learned from the FC layer and add a few more FC layers to adapt the new visual recognition task (Wu, 2017).

3.1.2.2.5 Dropout

Dropout was introduced by Srivastava in (Srivastava *et al.*, 2014), which proved very beneficial to the Neural Network training. The idea is to simply drop the information learned by certain neurons during the training process. During the training process, certain neurons might be relying on specific neurons of the previous layer. This suggests that the output is determined by only a few neurons if the training continues, however, the dropout randomly drops the information of such neurons. The dropping means to remove the neuron and stop the neurons from relying on certain specific neurons and avoid the overfitting problem. The parameter of Dropout is often represented by a value e.g., 0.5, which represents the removal of 50% of the neurons randomly during the training phase. A simple visualisation of the concept is given in Figure 20 (Picture Courtesy: (Zhu, 2019)).

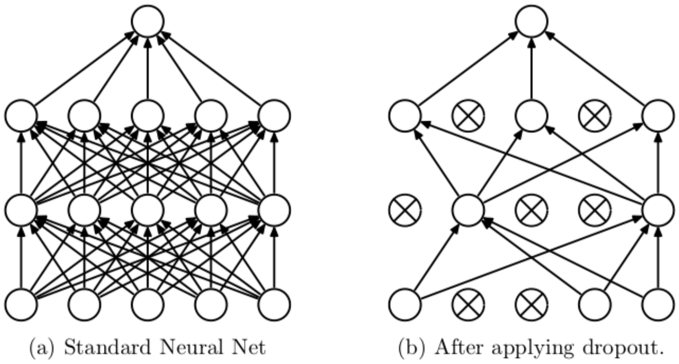


Figure 20A visual representation of what Dropout does to the neurons. As shown in the first figure that each neuron is connect with every single neuron from previous layer, however, the Dropout randomly removes the neurons and stops the Neural Network from relying.

Therefore, the Dropout allows the network to randomly omit half of the feature detectors to prevent complex co-adaptations on the training data and enhance generalization ability (Hinton *et al.*, 2012). Also, Dropout is an extremely effective method for ensemble learning (Baldi and Sadowski, 2013). A different version of Dropout is to drop the weights instead of feature detectors called DropConnect (Wan *et al.*, 2012). DropConnect proved more efficient and achieved better results but slower compared with networks with or without standard Dropout (Wager, Wang and Liang, 2013; Wang and Manning, 2013; Warde-farley *et al.*, 2014). Other generalization techniques are weight decay (Krizhevsky and Hinton, 2012), weight tying (Le *et al.*, 2010) or more are discussed in the work (Schmidhuber, 2014).

3.1.2.2.6 Batch Normalization

CNN training can be very complex. The inputs are distributed across each layer and they change continuously along with their associated weights. This can lead to a preference for certain inputs and make others ineffective during the training of the Neural Network. This means the Neural Network is unable to learn sufficient features and is biased during its training phase. To avoid preference and biased training, Batch Normalization was introduced. Batch Normalization is an operator added just before applying the activation function to normalize the inputs and scale these values for each layer. This normalization of the values allows Neural Network to achieve a higher learning rate, good initialization point, less resilience and avoid the issue of vanishing gradient (Ioffe and Szegedy, 2015)

3.1.2.2.7 Summary

Convolutional Neural Network has three main types of layers; Convolutional, Pooling and Fully Connected. A common practice is that the Convolution layer is followed by a Pooling Layer and ends with a Fully Connected Layer to give a predicted value required for the classification of an object at a Level Crossing. In general, CNN is trained in two main stages; forward and backward. Think of the forward network as a sequence of processing layers. The processing layers are arranged using the mentioned layers in different sequences depending on their application and progression in training and computational techniques. These layers would have a certain value of “weight” and “bias”. Some layers may not have any parameters to learn as well such as the Softmax layer as discussed earlier in the Softmax Function. These values along with a nonlinear function would give a certain output, which is an input for the next processing layer until it reaches the final layer's output. An additional layer for backpropagation (refer to Backpropagation) is added to the network that takes the final output and compares it with the ground-truth value. The last layer is a loss layer (refer to Loss Function), which compares the output value with the ground-truth value and adjusts the parameters accordingly. For the classification problem, the cross-entropy loss function as discussed in Cross-entropy Loss is often used to minimize the error. Once trained, the model is used to make a prediction where it is only run in the forward direction. The training of the network is stopped when it stops learning or the change in performance is insignificant. The parameters are optimized to minimize the error calculated during the training process. The comparison between the value of prediction and ground-truth value allows the network to change the parameters accordingly and then the model re-calculates these values in the forward direction. The new value would suggest how the Neural Network reacted to such a change and this measure of change is represented by a "partial derivation". The rate of change for this partial derivation is controlled by "learning rate", which is another parameter to control during the training process. For efficient and accurate results, the model should update parameters on each example but it would be time-consuming and computationally expensive. If the model is updated on a whole set of examples at the same time, it would be an inefficient approach, therefore, a compromise is made between calculating and updating parameters on the whole batch or each training example. This compromising technique is called Stochastic Gradient Descent-SGD, which takes mini-batches and takes averages on a certain number of training examples. For more details and mathematical explanation please refer to the work in (Kuo, 2016; Wu, 2017)The SGD approach allows the practitioner to reach a good average value that will minimize the error between predicted and actual ground-truth value, and this will update the model accordingly for training and deploying the network for its application. Once sufficient information is learned using kernels, the network is tested with some unseen images, for example, images from other sites of Level Crossings across Great Britain. The prediction of such images will evaluate the performance of the network. If the accuracy is enough for its application at a Level Crossing, the training process is stopped and the model is deployed at real-site.

3.1.2.2.8 Object Detection

Object Detection (Felzenszwalb *et al.*, 2010) is a process of not only classifying objects within a scene rather give the exact location of the detected object as well. This detection of an object provides a semantic understanding of the image and videos related to many different applications such as Image Classification (Krizhevsky and Hinton, 2012; Jia *et al.*, 2014), Face Recognition (Yang and Nevatia, 2016) and Autonomous Driving (Chen *et al.*, 2015). In recent years more attention has been paid to the Object Detection field, which deals with the challenges like a variation in poses, occlusions or light variations of an object along with the classification and localization (Girshick *et al.*, 2014; Ren, He and Girshick, 2015; Redmon *et al.*, 2016). The object detection model can localize and classify an object, hence the general pipeline of such models is Informative Region Selection, Feature Extraction and Classification as mentioned in the work (Zhao and Zheng, 2012).

3.1.2.2.8.1 Informative Region Selection

Since the desired object can appear anywhere in an image with varying position and aspect ratios, therefore, it is necessary to scan the whole image with multi-sliding windows. However, the multi-sliding windows are computationally extensive which could be avoided if the number of sliding windows are fixed but the fixed number of sliding windows may lose some information during the training process.

3.1.2.2.8.2 Feature Extraction

To have a robust and semantic understanding of the object, it is essential to gain a visual feature representation. Different manual feature extractor is available e.g., SIFT (Low, 2004), HOG (Dalal and Triggs, 2010) and Haar-Like (Lienhart and Maydt, 2002). But the real world is messy with different conditions in terms of light, background and appearances, therefore the manual feature extractor may not be suitable for its application.

3.1.2.3 Notable Models for Classification

In the end, a classifier is needed to distinguish the learned features and categories the objects accordingly. Different classifiers are used and this allows the network to be more semantic, informative and hierarchical for visual representation. Some good choices for classifiers are Support Vector Machine (Cortes and Vapnik, 1995), AdaBoost (Freund and Schapire, 1995) and Deformable Part-Based Model (DPM) (Felzenszwalb *et al.*, 2009).

Object Detection compared with simple Classification is to classify and then locate the object in an image. The location of the image is given by a rectangular bounding box, these bounding boxes represent the confidence of the network about the presence of that particular object in that particular position. For Object Detection, two different frameworks are widely used. One type of such framework is to generate “region proposals” and running it through an image and then classifying the objects, for example in R-CNN (Girshick *et al.*, 2014), SPP-net (He *et al.*, 2014) R-FCN (Dai *et al.*, 2016) or FPN (Lin *et al.*, 2017). The second type is to regard the detection of an object as a regression or classification problem, where a unified framework is used to give results, some common examples are MultiBox (Erhan *et al.*, 2014), YOLO (Redmon *et al.*, 2016) or DSSD (Fu *et al.*, 2017).

To understand the concept, it is essential to look into its novel implementations in the real world. Therefore, the novel and most commonly used image classification and Object Detection models are discussed. These models give insight into how basic layers are combined to design different architectures for different applications.

The basic layers of CNN architecture are used to make the earliest simple Neural Network to a more complex and denser Neural Network. To understand how these layers are compiled together to design a CNN architecture from scratch, it is necessary to discuss some of the notable models used for image classification.

3.1.2.3.1 AlexNet

The work in (Krizhevsky and Hinton, 2012) competes in the “ImageNet Large Scale Visual Recognition Challenge '' in 2012, where the model achieves a top-5 error of 15.3% and gained the first position. The work introduces the use of ReLU instead of other non-linearity functions e.g., Tanh, which accelerates the speed by 6 times compared with the accuracy from other nonlinear functions. Dropout was another concept used in this work for the regularization technique, which is simply dropping weights randomly to avoid overfitting. The model did perform well with this technique but the training time is significantly increased because of the Dropout technique. The model uses millions of images from ImageNet which has 62.3 million parameters to train and is achieved using two GPUs. The AlexNet consists of 8 learned layers; 5 Convolution Layers and 3 Fully Connected Layers. Figure 39 in 10.1 gives detailed information about the AlexNet including the input and output layers along with the number of parameters during the training process.

3.1.2.3.2 VGG-Net

The work in (Simonyan and Zisserman, 2015) proposed a VGG-Net, which uses small convolution filters of size 3*3 as compared to the large size filters in AlexNet of size 11*11. The use of small filters with a small stride value allows the network to use more ReLU units, which increases the nonlinearity and makes the network more discriminative. These small kernels allow the network to have more layers, where it can learn more complex features, hence the accuracy is further improved. The VGG-Net achieved a top-5 error of 6.8% in the "ImageNet Large Scale Visual Recognition Challenge" competition in 2014. Different versions of the VGG-Net are available and they only differ in their number of layers; VGG-16 or VGG-19 etc. Figure 40 in 10.2 gives detailed information about VGG-Net.

3.1.2.3.3 Inception

AlexNet mostly uses large kernels, which are responsible to learn more global features. The VGG-Net can use a "fixed sized-kernel" which learns more distributed and area-specific features across the image. The Inception (Szegedy *et al.*, 2015) does not worry about the depth of the network, rather the main concern is the width of the network, where it uses different size kernels to learn both global and distributed features to select the most appropriate ones for the training process. The process of using different size kernels and learning features from all of these kernels is called the "Inception module". The Inception network achieved a top-5 error of 6.67% on the "ImageNet Large Scale Visual Recognition Challenge" competition in 2014. Figure 41 in 10.3 shows a detailed structure of the Inception network along with its "modules".

3.1.2.3.4 Xception

Work in (Chollet, 2017) proposed a CNN architecture based entirely on depthwise separable convolution layers. The work introduces a hypothesis that "*mapping of cross-channel correlations and spatial correlations in the feature maps of convolutional neural networks can be entirely decoupled*". The mentioned hypothesis is an Extreme version of Inception, hence the name "Xception".

3.1.2.3.5 DenseNet

Convolutional Neural Network comprises of layers, where each layer implements a non-linear transformation and a composite function such as Batch Normalization, Pooling or Rectified Linear Units (ReLU). The work in (Huang *et al.*, 2017) proposed the idea of each layer connected by all subsequent layers in the network. Therefore, the layers receive feature maps from all preceding layers, hence the name "Dense". The architecture and network details are mentioned in the work (Huang *et al.*, 2017).

3.1.2.3.6 Summary

A common thing among the aforementioned famous architecture is the basic layers used to make the architecture of the CNN network. The Convolution layer is followed by the ReLU function to add non-linearity followed by Pooling Layer and ending with Fully-Connected Layer. These layers and their combination are used to design an architecture for an Image Classification network. These famous architectures give a clear insight into how these basic layers perform so well in their application and achieve high accuracy. A brief comparison between these discussed architectures is given below in Table 5.

Network	Year	Top1 Accuracy (ImageNet)	Top5 Accuracy (ImageNet)	Parameters- M
AlexNet	2012	63.3%	84.6%	60
VGGNet-19 ***VGGNet- 16***	2014	74.5% 74.4%	92.0% 91.9%	144 138
InceptionV1	2014	69.8%	89.9%	5
Xception	2016	79%	94.5%	22.8
DenseNet-121	2016	74.98%	92.29%	-

Table 5A brief comparison of the discussed networks. Their top5 and top1 accuracy on the ImageNet benchmark are mentioned as well along with their parameters.

3.1.2.4 Notable Models for Object Detection

The RCNN model and YOLO model is selected because several sub-versions of such models are available, which will provide sufficient information for the reader to understand the difference and importance of each approach available for its application in the detection of an object.

3.1.2.4.1 R-CNN Models

The R-CNN network (Girshick *et al.*, 2014), which may stand for “Region Convolutional Neural Network” or “Region-Based Convolutional Neural Network” is the most commonly used model to adapt the “region proposal” approach. The authors of the R-CNN network proposed different techniques to improve the downside of the previous version. The models include; RCNN, Fast-RCNN and Faster-RCNN.

3.1.2.4.1.1 R-CNN

Within the application of Object Detection, Localization and Classification, the R-CNN network is considered as the first large and successful model to achieve state-of-the-art results on benchmark datasets e.g., VOC-2012 and 200-class ILSVRC-2013. The model architecture is shown in Figure 21

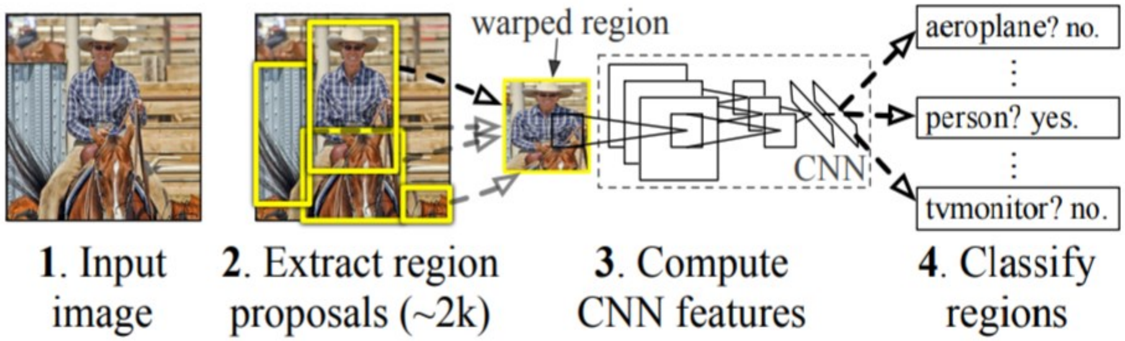


Figure 21The architecture of the R-CNN model. The model divides itself into four sub-stages; Input Image, Region Proposals, CNN Features and Classification.

The model is subdivided into three main stages

3.1.2.4.1.1.1 Region Proposal

Computer Vision algorithms are used to generate region proposals, which are category independent. These region proposals are called candidate region proposals, which are extracted for the second stage. The R–CNN models extract 2000 region proposals during this stage. (Note: Due to the flexible network, it is possible to use a different algorithm for generating and extracting Region Proposals.)

3.1.2.4.1.1.2 Feature Extractor

A Convolutional Neural Network, which in this case is AlexNet (Krizhevsky and Hinton, 2012) (discussed here AlexNet) is used as a feature extractor. The feature extractor extracts 4096 element vectors from these proposed regions, which describes the contents of the image in that particular region.

3.1.2.4.1.1.3 Classifier

A classifier e.g., Support Vector Machine-SVM (as discussed in Support Vector Machine) is used to classify the object from the features learned by the CNN model on these extracted regions. One type of SVM is trained for one particular category.

The downside to this approach is that it is computationally expensive as it requires the model to generate and extract regions. Also, the time the CNN requires to extract features from a given region proposal makes the model slow. This effect the functionality and reliability of the network in a real-time application as required in a Level Crossing site.

3.1.2.4.1.2 Fast-RCNN

To overcome the issue of speed, the work in (Girshick, 2015) proposed a single model of architecture compared with three stages in the R-CNN model. The working principle of Fast-RCNN architecture is given in Figure 23

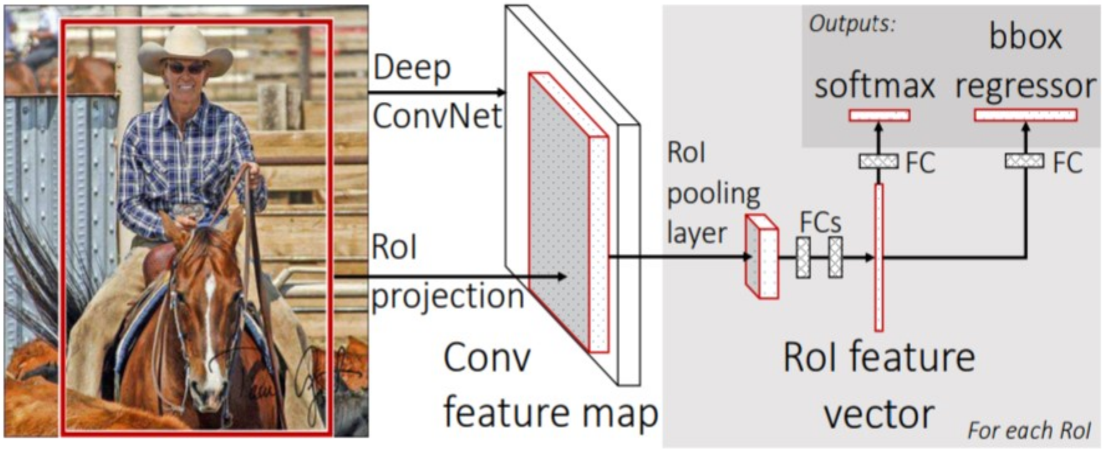


Figure 22A summary of Fast-RCN model, where the whole architecture is not divided into stages rather works as one single model.

The model takes a set of "region proposals" from a Deep CNN network, where a pre-trained Convolutional Neural Network e.g., VGG-16 (as discussed here VGG-Net) is used to extract features. The features are fed to the last custom layer of the CNN network called "Region of Interest-RoI Pooling Layer". The RoI Pooling Layer extracts specific features, which belongs to the specific input candidate region. Once these features are extracted, the fully connected layer of the network gives two outputs; one output from the Softmax layer, which categorizes the object and the other output is from the bbox regressor, which gives the bounding box for the detected object. The model is quicker in the training and prediction phase when compared with the first proposed RCNN model, however, it still requires a set of candidate regions for each image.

3.1.2.4.1.3 Faster RCNN

To further improve the limitations given in the R-CNN and Fast-RCNN model, the Faster RCNN (Ren, He and Girshick, 2015) was proposed. The Faster-RCNN achieved the first-place results in two competition tasks for Object Detection; ILSVRC-2015 and MS COCO-2015. The paper discusses the improvements in these words

> *"… our detection system has a frame rate of 5fps (including all steps) on a GPU while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. In ILSVRC and COCO 2015 competitions, Faster R-CNN and RPN are the foundations of the 1st-place winning entries in several tracks"*

The model architecture is shown in Figure 23 (Picture Courtesy: (Tsang, 2018)).

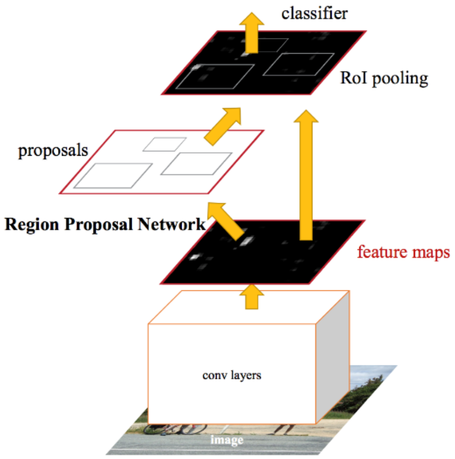


Figure 23A overview of Faster-RCNN network. The model is fed with proposal regions, which is then fed to the classifier for predictions.

The Faster-RCNN model uses a Convolution Neural Network to propose multiple regions to consider. The feature extractor then extracts features from the proposed regions to predict the class and bounding box for each category. This reduces the time and unnecessary computational power required for generating and extracting region proposals using different techniques or stages. The Faster-RCNN is most efficient and achieves state-of-the-art results within Object Recognition tasks.

The RCNN models are a prime example of the Object Detection framework, where the model uses Region Proposals to detect, localise and classify objects. These models, particularly the Faster RCNN, have high accuracy and work efficiently for Object Detection applications.

Other models regard the problem as regression or classification, where multiple models are used. The following work will consider one such model called the YOLO network along with its improved versions.

3.1.2.4.2 YOLO Models

The YOLO model "You Only Look Once" is proposed in (Redmon *et al.*, 2016), where the model is regarded as one single Neural Network trained end-to-end for prediction. The YOLO network cares more about the speed compared with the accuracy of the network as described in their work in the following words

"Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second …"

The model working principle is shown in Figure 24

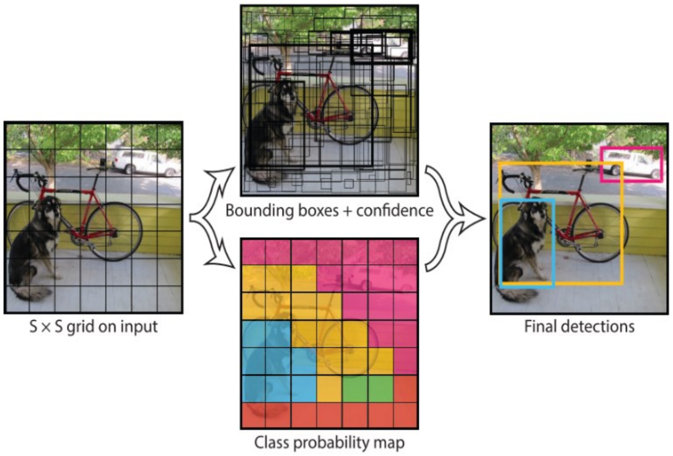


Figure 24YOLO working principle summarised, where an input image is divided into multiple grid cell to give predictions.

The YOLO network divides an image into multiple grids. Each cell grid would predict a bounding box. The bounding box will give x and y coordinates along with its class and confidence metrics. These outputs e.g., bounding boxes and class along with their probability are combined to final sets of bounding boxes and class probabilities. This approach is much faster than the RCNN models but the localisation coordinates of predicted objects are not as accurate compared with the RCNN models.

3.1.2.4.2.1 YOLO v2 and YOLO v3

To further improve the performance of the YOLO network, the work proposed in (Redmon and Farhadi, 2017) make use of predefined anchor boxes. These predefined bounding boxes come in different sizes and shapes, which are tailored during the training process. The choice of these bounding boxes is determined using k-means analysis on the training dataset. The model is more stable to small changes and works better compared to its previous versions. The width and height of the bounding boxes are predicted as an offset from the cluster centroids and the coordinates are predicted relative to the location of filter application using a sigmoid function as shown in Figure 25

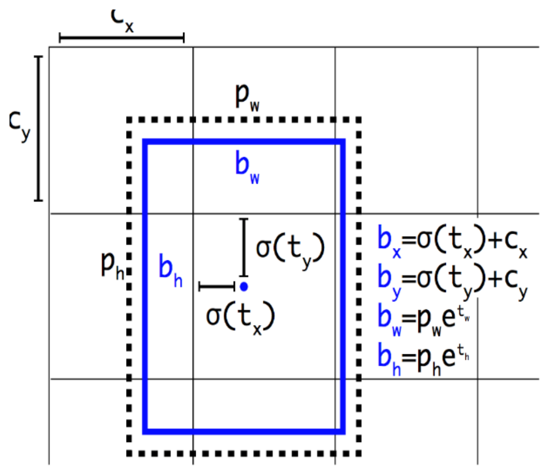


Figure 25The bounding boxes are predicted from the cluster centroid and the coordinates from a sigmoid function for YOLO vs model.

Small changes were made to the YOLO v2 model and proposed in (Redmon and Farhadi, 2018). These changes allow the model to make more real-time predictions at a faster rate. For more detailed metrics and comparisons between models please refer to the paper in (Redmon and Farhadi, 2018).

3.1.2.4.3 MobileNet

MobileNet (Howard *et al.*, 2017) is a lightweight architecture with low maintenance thus improving the performance and speed of the model. MobileNet uses depth-wise separable convolutions, which means the model applies a single filter to each input channel. The pointwise convolution then applies 1*1 convolution to combine the outputs of depthwise convolution. This has been clearly explained in the paper as “A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.” The overall architecture of MobileNet has 30 layers consisting of convolution, dept wise and pointwise layers. MobileNet uses depth-wise separable convolution to reduce model size and complexity particularly useful for mobile and embedded vision applications.

3.1.2.4.4 ResNet

Earlier models rely on the concept of “stacking layers” and making a network deeper. This allows the network to learn more feature maps but has the risk of a “vanishing gradient”. The work in (Lin *et al.*, 2017) argues that stacking “identity mappings” upon the network would still allow the deep network to achieve high results and perform the same without being affected by the “vanishing gradient” problem. The ResNet achieved the highest accuracy in the LSVRC2012 challenge after AlexNet.

3.1.2.4.5 Efficient-D7

Efficient-D7 (Tan, Pang and Le, 2020) was built on the idea that efficient Object Detectors should achieve high accuracy with low computational power. Every Object Detector has three main components; a backbone to extract features, a featured network that takes multiple levels of features to represent salient characteristics of an image and a class/box network for prediction. Efficient-D7 used the Efficient model (Tan and Le, 2019) for its backbone network, which is more powerful and efficient compared with other traditional networks e.g., ResNet or AmoebaNet. The traditional feature networks like top-down Feature Pyramid Network (FPN) or complex feature networks like NAS-FPN, which include the bottom-up flow of information as well and is replaced with new bi-directional feature network Bi-FPN. Bi-directional Feature Network allows a more efficient flow of information in both directions; top-bottom and bottom-top. Another key feature in Efficient-D7 is the addition of weight value for each input feature and replacing regular convolution with depthwise separable convolutions. This change allows the network to "improves the accuracy by 4% while reducing the computation cost by 50%." The model also uses the compound scaling method for its depth/width/resolution, where one single compound scaling will control all these dimensions. Efficient-D7 achieves a mean average precision (mAP) of 52.2 achieving state-of-the-art results. For more details on the model's architecture and discussion refer to their paper (Tan, Pang and Le, 2020).

3.1.2.4.6 Summary

More details on their particular results and how they can be deployed at a Level Crossings site is discussed later. A summary of each discussed model along with their accuracy and detection speed is mentioned in Table 5.

Model	Speed (ms)	COCO mAP	Outputs
Efficient-D7	325	51.2	Boxes
MobileNet 320*320 **MobileNet 640*640**	19 48	20.2 29.1	Boxes
ResNet50	46	34.4	Boxes
YOLO	-	51.5	Boxes

Table 6A summary of detection models, where the model's Speed and mean Average Precision (mAP) are mentioned. The mAP values are based on COCO evaluation metrics.

The discussed models and their accuracy do not mean that they are applicable at a Level Crossing. A common approach is to train different notable models using the same dataset and choose one particular model with the highest accuracy. Once the model is selected, the model's accuracy is improved using optimization techniques. The model selected for the Level Crossing application requires speed, reliability and stability, which should perform efficiently in real-time. It should predict at a rate of 15fps, where a high rate of prediction would indicate that even if at some instance the model incorrectly predicts a category at a Level Crossings it will be replaced by many other positive predictions before a human can notice such a failure. This will ensure a fail-safe mechanism of the Interlocking system with SIL 3 or SIL 4 standard.

The most important part of any Machine Learning project is to collect the right dataset for the model. This section will briefly mention the datasets used to train and evaluate the models for Image Classification and Detection using Images from Cameras and micro-Doppler signals from the RADAR for its application at a Level Crossing.

3.2 DATASET- CAMERA

The ImageNet dataset is an open-source platform (ImageNet, 2020), which provides over 15 million high-resolution labelled images with around 22 categories for classification. The required application for a Level Crossing should classify and detect particular objects, for example, a pedestrian (adult and child), bicycles and vehicles. Therefore, the dataset downloaded from the ImageNet should be of these particular categories. These images are arranged in tree-directory format, where the folder name represents the label for images within that particular folder. These images are divided into “train” and “validation” datasets. The training dataset is used for training the Neural Network and Validation dataset is used to evaluate the performance of such Networks during the training. The Validation helps the practitioner to avoid Overfitting. Sometimes another dataset called “test” is used to evaluate the training and validation processes. Accuracy and Losses metrics are used to compare the results and evaluate the model’s performance. Often the downloaded images from open-source are not very specific to their given label and may represent another object with more dominant features. For example, the dataset downloaded for pedestrians (adult and child) contains vehicles or bicycles within a major part of an image, which disrupts the feature learning process and makes the model ineffective. Therefore, the downloaded images are manually checked and any image with a strong dominance of other objects are excluded from the dataset. However, the dataset may still contain some images where an image represents another object and a small error is still expected. Excluding images from the dataset may reduce the diversity of the given categories. To overcome the issue of diversity, data augmentation techniques are used to add images to the dataset as shown in Figure 29. The images in Figure 29 represent images after certain operations e.g., rotation, shift, zoom and flip are applied to particular images. To completely overcome the issue of diversity within the image dataset and bias learning the dataset should contain images with strictly one object for the given category (which is time-consuming and less efficient) or the images should be from the same distribution e.g., Level Crossing. The collection of datasets from the same distribution e.g., Level Crossing is discussed in DATASET improvement and expansion. The new dataset collected with the cooperation of relevant authorities will strongly correlate with the situation at the time of the model’s interference. To evaluate the Image Classifier, the model’s accuracy on training and validation dataset is used. Their relevant details e.g., trainable parameters are mentioned as well. The accuracy and loss metrics will give a fair comparison since the dataset and augmentations techniques used to train the models are kept constant for each given model.

Object Detection is the right progression from Image Classification, where Object Detection gives two outputs; classification and bounding boxes for the precise location of an object. To train a model for Object Detection, the dataset requires specific values around the relevant objects e.g., bounding boxes. These values correspond to the coordinates of a rectangle around each object within an image. Some models within the Object Detection family require these values in a *text* file, while other models require *XML* format. The format depends on what particular model of Object Detection is used for training. Most models in Object Detection requires *XML* files, whereas, other models e.g., YOLO works on *text* file as well. These files are generated using many different annotations tools. Initially, the dataset is prepared using “*LabelImg”*. “*LabelImg”* allows the user to annotate the objects using rectangular boxes, where each annotation will generate the relevant *XML* file. Generally, the model trained using Transfer Learning techniques requires about 200 or more images for each given category. The process to manually annotate images and generate relevant files is slow. To generate a large number of images more efficiently, the OIDv4 toolkit is a powerful tool that allows users to download custom objects from a large open-source “Open Image Dataset”. Open Image Dataset has about 1M images containing about 600 classes. OIDv4 toolkit allows the user to download objects belonging to a specific category along with their relevant files. Using the OIDv4 toolkit, the new dataset prepared for YOLO are 2000 images for each category in the training dataset and 200 images in the test dataset. To evaluate the Object Detector, the model's *Loss* metrics are used to compare the results from each given model. Some data augmentation and preprocessing techniques are used to add diversity to the dataset e.g., noise, blur, grayscale and resize operations. The diverse dataset is shown in Figure 37. The *Loss* metrics are compared since other hyperparameters and data augmentation techniques are constant for each given model. To visualize the results relevant graphs are mentioned in Object Detection using Transfer Learning techniques, which are generated using a tensor board.

3.3 DATASET- RADAR

RADAR is used in different applications depending on its post-processing techniques and the type of RADAR used. Traditional approaches for RADAR use simple post-processing techniques to calculate the speed, range or direction. For example, a simple RADAR would transmit a pulse, which is reflected if an obstacle was present in its direction of propagation. The distance between the transmitted and received signal would determine the distance between two objects (Wolff, 2020). RADAR imaging is another interesting area of research, where the aim is to recognize objects using different information obtained from the RADAR system. Synthetic Aperture RADAR- SAR is commonly used to obtain RADAR images, which are obtained from Doppler frequency, which is different from Optical pixel Images. The SAR image resolution compared with image pixels are independent of range and can operate regardless of long-range and cloudy conditions (Skolnik, 2020). The present work is not to discuss the technical aspect of RADAR since they are already installed at a Level Crossing site and for a more detailed discussion on the RADAR system please refer to the course offered by Robert O'Donnell (O'Donnell, 2007).

The Deep Learning approach in this work uses micro-Doppler signals (Chen, 2011) from RADAR to identify objects. These micro-Doppler signals are generated from a small movement of objects e.g., a swing of an arm for pedestrians or movement of pedals for a bicyclist. MATLAB (Matlab, 2020a) has in-built functions and a toolbox to stimulate short-term Fourier Transform (STFT) of radar signals, which in turn is used to generate micro-Doppler signals. These micro-Doppler signals are distinguished for each category depending on its signal generation as shown in Figure 39. These micro-Doppler signals are used as an input for the Convolutional Neural Network to learn representation from RADAR signals. MATLAB allows users to generate a diverse dataset required for the CNN model to learn sufficient information for classification. The diversity within a dataset includes variations in speed, range and direction. It also includes noise signals and overlapping of an object to resemble a real-work scenario.

For the proposed work, the model generates 200 Radar signals for each given category to generate micro-Doppler signals used for classification. These signals are generated using a predefined set of properties for each given model. For pedestrians, the model randomly uses a height between 1.5 and 2 meters with a speed variation from 0m/s to 1.4*H m/s, where H is the height used. For Bicyclist and Car signals the model randomly uses a value between 0m/s to 10m/s. These conditions perfectly resemble the dynamic environment of the Level Crossing site. The generated dataset is then open sourced for future research and contributions. To further improve the accuracy, more signals should be generated directly from the RADAR rather than simulated using a toolbox available from MATLAB. The generated signals would correlate to the actual environment using the actual type of RADAR used. However, the present work uses a simulated dataset with a limited dataset because of the computational limitation and inability to access a physical site. (The present work was done during the pandemic situation in the world because of COVID19 in 2020).

4 Deep Learning For Level Crossing

The present section will discuss the application of Deep Learning at Level Crossing. It divides the application into three different stages: Classification using CCTV, Detection using CCTV, Classification using RADAR. The training process, models and their results are discussed in detail for each stage of its application.

4.1 Image Classification using CCTV

For application in Image Classification, two main approaches are utilised; Training from Scratch and Transfer Learning. Training from scratch requires a large number of the labelled dataset, defining the architecture and optimising the hyper-parameters for the given architecture along with other expertise required during the training process e.g., evaluating the losses and determining the best possible solution for a particular application. The process does not have any definite rules for higher accuracy or defining an architecture, however, some common practises and procedures are adopted by experts within the field of Computer Vision, which have been beneficial for many other applications. The Transfer Learning (or feature extractor) approach utilises the architecture of some notable models within Computer Vision applications (some are discussed in Image Classification-Models). Along with the given architecture of the model, the approach also uses the "weights" of these pre-trained models. These models have been trained on millions of images for thousands of categories and the "spatial sharing" and "parameter sharing" features of CNN would allow researchers to train their model on a small set of training images without worrying about the architecture itself. The proposed work utilises both of these approaches for comparative study before finalising a model for its application at a Level Crossing.

4.1.1 Training From Scratch

As outlined in the work of Andrew Ng. Machine Learning Yearning (Ng, 2016), the most important part of the CNN model is its dataset.

1. The dataset should be sufficient enough to learn representation from the dataset for Image Classification.
2. The dataset should be from the same distribution e.g., Level Crossing to avoid any bias.

The dataset should be divided in the right proportion for training, validation and testing. A general rule is to have about 90-95% of the dataset for training and the remaining for validation and test. The validation is just another way to avoid overfitting in the test dataset during the trial-and-error method of training the CNN model.

The dataset used for the application of Classification at a Level Crossing consists of 2000 images for each category; vehicle, bicycle, pedestrian and child. The same number of images will ensure the CNN does not prefer one particular category by learning strong features of one and leaving other categories. Most of these images are from the same distribution e.g., Level Crossing's site, which is again to avoid any bias. Some of these labelled images are shown in Figure 26. As shown in Figure 26, the images are collected from different sites and types of Level Crossings. These sites and types of Level Crossings reflect the actual site of application; hence the same distribution of dataset would achieve better results.

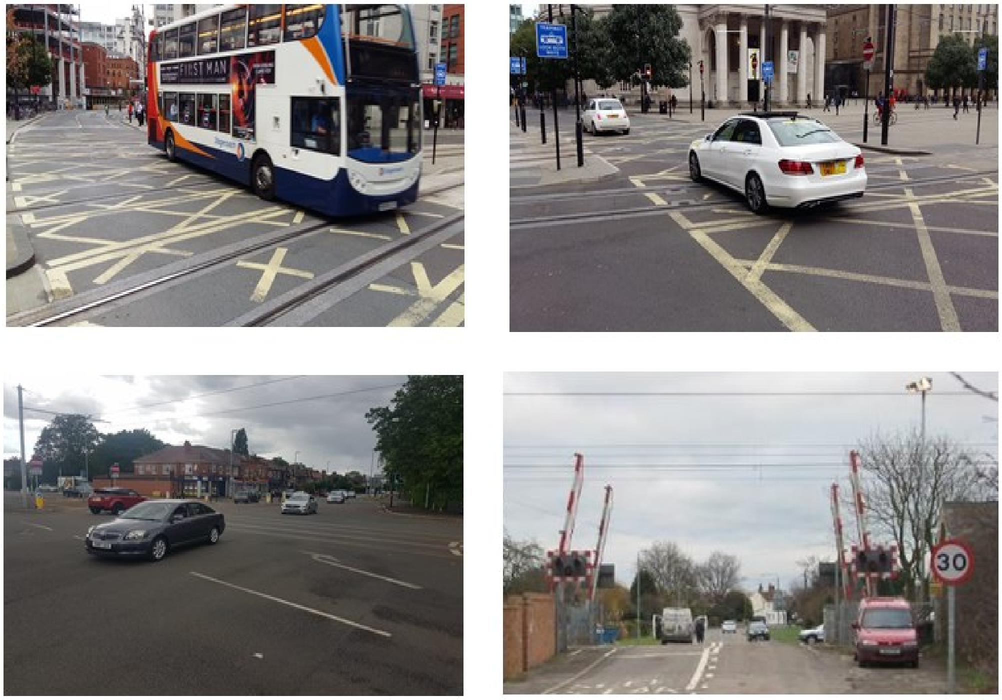


Figure 26Representation of the dataset used for Image Classification. Images are collected from different sites and types of Level Crossing to add diversity in training process.

These labelled images are stored into relevant groups; train, validation and test sets. These images are then ready to be fed into a Neural Network. As mentioned earlier, there are no definite rules to determine the perfect architecture of a Neural Network, which will achieve the highest accuracy. Hence a common approach to define and optimize the hyperparameters is achieved as mentioned in (Ng, 2018). To understand the importance of the "Convolution" layer, firstly, a traditional Neural Network is defined using only the "Dense" layer. The details of the given Neural Network are mentioned in Table 7.

	Architecture	**Trainable Parameters**	**Accuracy**
Traditional Neural Network	Input Layer Flatten Layer Dense Layer (2x)	34M	Training acc: 53.84% Validation acc: 41.75%

Table 7Traditional Neural Network with just one Input and Flatten Layer and two Dense Layer with 34M parameter.

The simple architecture as mentioned in Table 7 consists of only Dense layers with no "Convolution" layers. The model achieved an accuracy of about 53.84% for the training dataset and 41.75% for the validation dataset. The reason for the difference is because the validation dataset is "foreign" to the model during the training stage. The model can make predictions on the validation dataset using the learned "features" from the training dataset. A "Convolution" layer is added to the architecture, which will give an idea of how a Convolution layer affects the performance of the Neural Network. The details of the new architecture are given in Table 8.

	Architecture	Trainable Parameters	Accuracy
Convolutional Neural Network	Input Layer Convolutional Layer Max Pooling Layer Flatten Layer Dense Layer (2x)	44M	Training acc: 99.47% Validation acc: 52.05%

Table 8Convolutional Neural Network with only one Convolution Layer. The architecture, trainable parameters and accuracy are mentioned. The trainable parameter increased from 34M to 44M.

With one "Convolution" layer the accuracy on the training dataset increased from about 53.84% to 99.47%, which is roughly an increase of 46%. The main reason for such an increase is its ability to learn representations from the dataset. The trainable parameters increased from 34M to 44M, an increase of 11M with just one Convolution Layer. The new parameters learn representations from the dataset and correlate with the category it belongs. The visualization of the learned features or representation from Convolution layers is often used to show the operation of a Convolution Layer (shown later with multiple layers on Convolution). The results look near-perfect and highly accurate but that is a misinterpretation of such values. To understand the given results more accurately, let us plot the given results as shown in Figure 27.

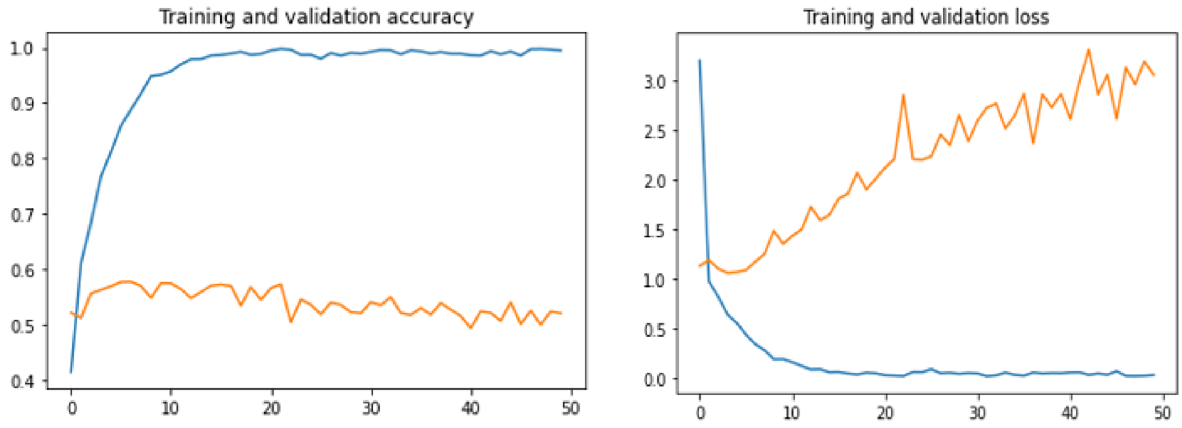


Figure 27Training vs. Validation plot of accuracy and loss for given number of epochs (e.g. 50).

As shown in Figure 27, the validation and training values (both accuracy and loss) have a large difference. The model performed well with the training dataset and achieved almost a perfect model of 100% accuracy. Comparatively, it performed poorly with the validation dataset and achieved an accuracy of about 50%. The model is said to be "Overfitting". Overfitting is achieved when the model performs well on training but poorly on unseen datasets e.g., validation or test datasets. This is because the model has learned enough representation to perfectly correlate all the training data to its corresponding labels, hence it achieves an accuracy of almost 100%. However, the same model when tested with the unseen data e.g., validation dataset, the network was unable to classify the images correctly rather it gave every other prediction wrong. To avoid such a situation, the model should be fed with more diversity of images in its training dataset. The diversity of these images will allow the network to learn more abstract features or representations from the dataset so it can classify the unseen data more accurately. However, it is not always possible to gather more datasets regardless of how many open-source image datasets are available. There are two main reasons for such limitations

1. The opensource images are rough representations of the category e.g., vehicle or pedestrian. These images if downloaded from different opensource may have many similar images hence increasing bias to the network. This will further demand the cleaning and filtration of these images avoiding similar images or irrelevant images.
2. The open-source image dataset is not necessarily from the same distribution as it is required for its successful application. Therefore, the network may perform well with the given dataset but poorly when implemented for its application at a Level Crossing.

For such particular reasons, the addition of a dataset may not always be a preferable solution for the “Overfitting” problem. Therefore, researchers have introduced many “Data-Augmentation” techniques. These techniques introduce variations in the dataset without really adding other images. The most common “Data Augmentation” techniques are Rotation, width or height shift, Zoom and horizontal flip. The technique's names indicate the effects or “operation” it performs on an image. These techniques are used to add variation and diversity to our training dataset to avoid the overfitting problem. Figure 28 shows images after these techniques have been applied to our image dataset.

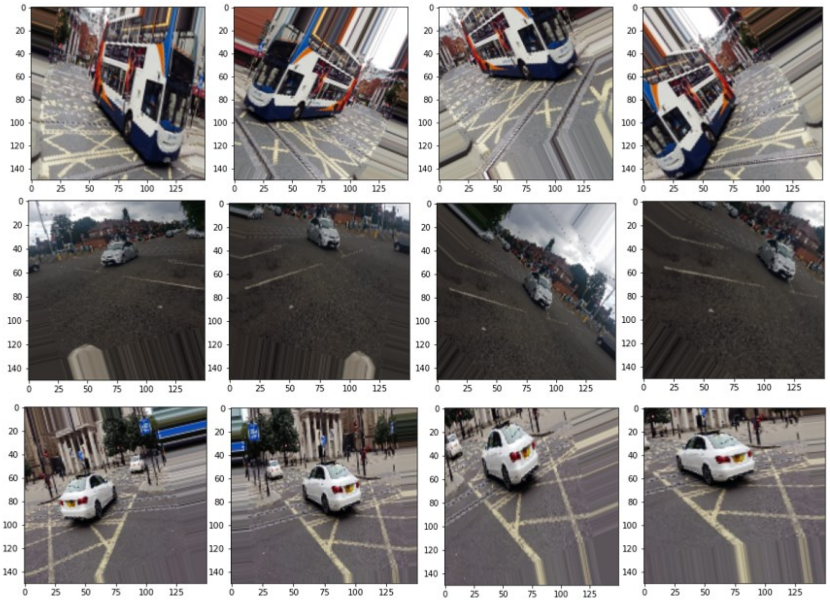


Figure 28Data Augmentation on three different images from our training dataset. These images have been zoomed, shifted and flipped as shown.

A common practice is to add a “Dropout” layer along with some data augmented techniques to avoid overfitting problems. As discussed in Convolutional Neural Network, the dropout is randomly dropping the weights during the training process and it does improve the accuracy of the network. The final model architecture along with its relevant details are mentioned in Table 9.

	Architecture	**Trainable Parameters**	**Accuracy**
Convolutional Neural Network	Input Layer Convolution Layer & Max Pooling Layer (4x) Flatten Layer Dense Layer Dropout Layer Dense Layer	3M	Training acc: 66.78% Validation acc: 69.63%

Table 9A Convolutional Neural Network with more layers e.g., Convolutional Layers and Pooling Layers. It has less trainable parameters because of Dropout Layer.

As shown in Table 9, the trainable parameters significantly decreased from 44M to just about 3M mainly because of the Dropout Layer. The Dropout Layer randomly drops the trainable weights to avoid the Overfitting problem.

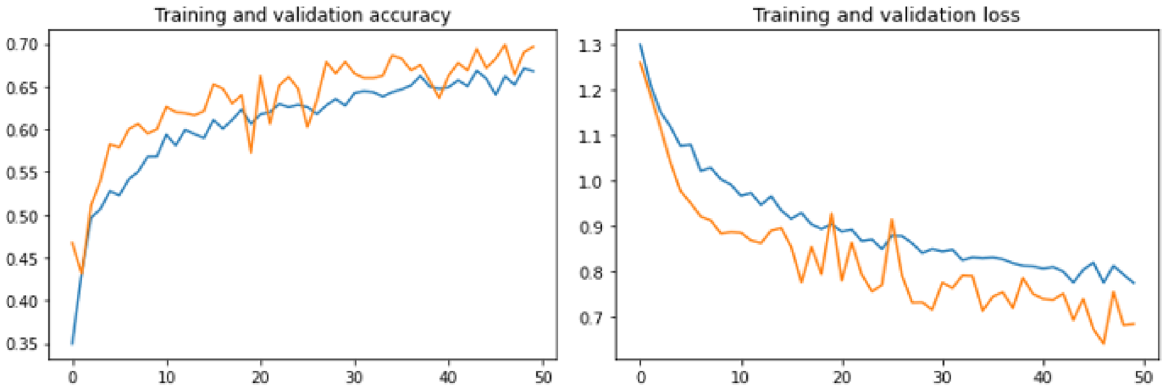


Figure 29The training and validation accuracy and loss graph after data-augmentation techniques and addition of dropout layer.

Figure 29 clearly shows that the model is not “Overfitting” rather it strongly correlates with the accuracy of the validation dataset. The accuracy of the training dataset decreased from 99% to 66.78% and validation accuracy increased from 52.05% to 69.63%, which is roughly a 20% increase. The accuracy of such a model is improved with two approaches which could be visualised as improving in a vertical or horizontal direction.

1. *Vertical:* The depth of the network is increased by adding more layers e.g., Convolution Layers or Pooling Layers. This will allow the network to learn more representations from the available layers and achieve better results. As mentioned earlier, the suggestion of introducing new layers is not a definite proposal for improving the accuracy of the network. Sometimes the network has learned enough representation from the dataset given that it can no longer learn new features regardless of new layers in the network. The network may have reached its maximum capacity for its particular application.
2. *Horizontal:* Instead of adding more layers, the network is fed with more data or the hyperparameters are optimised to increase its ability to learn more representations. Data Augmentation techniques

along with Dropout Layer are such practices. Often manual interference is required to analyse the images available for training to filter out any anomaly and select the images with the most relevance with its application. Sometimes the number of filters or their sizes along with other parameters are tuned before deciding which particular approach improved the accuracy of the network.

The mentioned reasons are few among many other things that should be considered when designing a CNN from scratch and it is an intensive task that requires expertise, dataset, time and high computational power. Since the scope of the proposed work is not designing an architecture rather proposing one for its application at a Level Crossing, we shall suffice ourselves with the given brief introduction on CNN from scratch with an accuracy of about 70%. For most computer vision applications, the researchers utilise the models and weights available from pre-trained Neural Networks. These models have been trained on millions of images and achieved high accuracy. The availability of these networks helps the researcher in two most significant ways

1. The pre-defined architecture is available which have been trained on millions of images achieving state-of-the-art results in Computer Vision applications. Therefore, the struggle to define a perfect architecture using trial and error and relevant expertise is avoided when the end target is not architecture rather its application. Different notable works and their architectures are available for Feature Extraction or Transfer Learning and not necessarily every model will perform well, therefore, it is good practice to run your dataset with few notable works and analyse the results before choosing one for post-processing and its application.
2. The architecture does help the researchers with the structure of the Neural Network; however, it does not solve the problem of the limited dataset. Because the limited dataset may still perform poorly regardless of the number of Convolutional Layers. To overcome the issue of the limited dataset, the authors of these notable architectures have open-sourced the “weights” as well. These weights contain the trained “weights values” of learned features and representations from their millions of images. These layers can retain the abstract and common features e.g., edges or lines. From these learned features, it is comparatively easy for Neural Network to learn motifs and higher-level features belonging to a particular category.

Therefore, the Neural Network from scratch is not continued; rather a Feature Extraction or Transfer Learning approach is used for the Level Crossing application. Before continuing with Transfer Learning, the learned representation from the multiple Convolution Layers from the given model in Table 8 is visualised to see its operation and demonstrate the impact of Convolution Layer compared with Traditional Neural Network. Two different types of vehicles are fed to the trained network for some pre-processing and giving visual representations of the learned features.

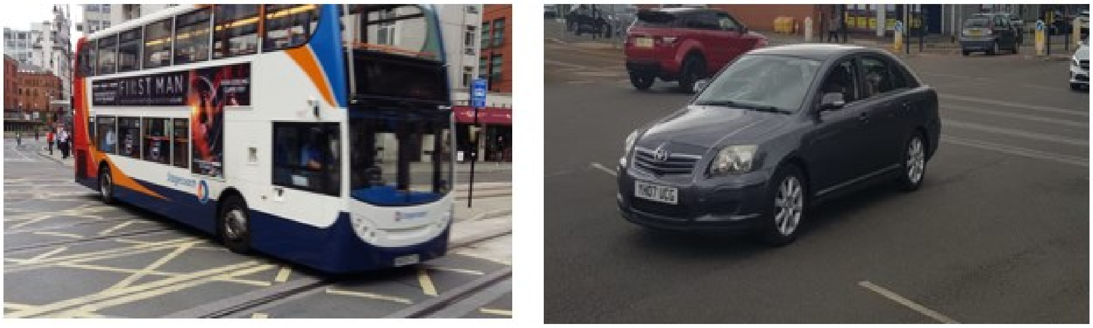


Figure 30Two images at random are selected from different sites of a Level Crossing area. These images are used to visualize the representations learned from the Convolution layers.

Figure 30 shows two images, which represents different sites of a Level Crossing area. The sites are dynamic backgrounds and different textures of ground. Such variations and diversity are an ideal situation where the proposed system must learn and correctly classify the objects at a Level Crossing.

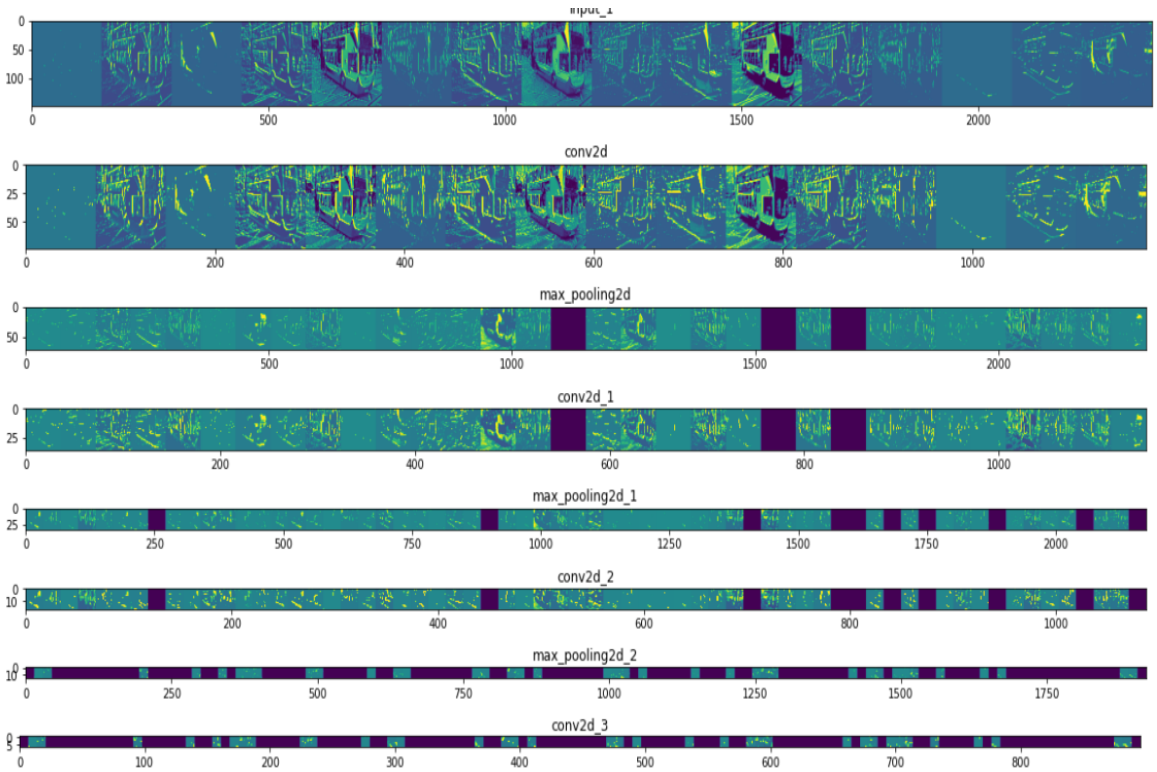


Figure 31Visual representation of the bus at a Level Crossing site. This visual representation represents the features learned from successive Convolution Layers within a Neural Network.

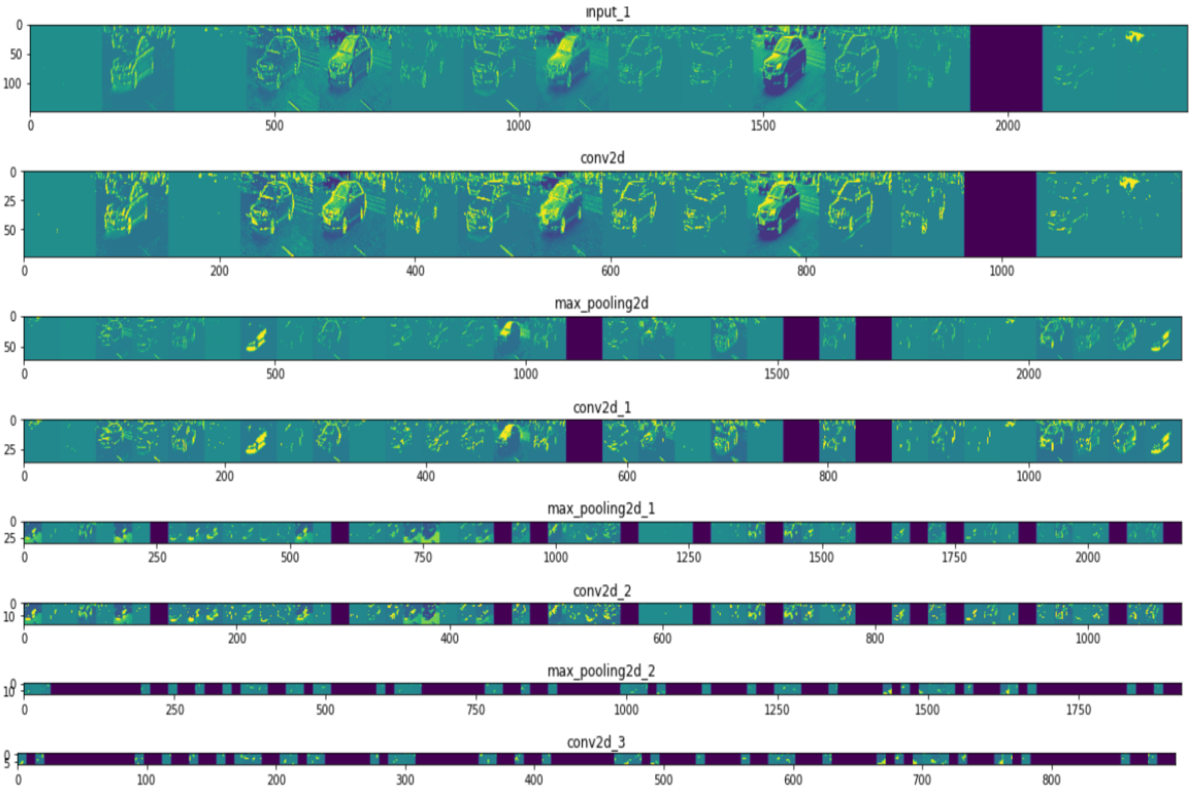


Figure 32Visual representation of the car at a Level Crossing site. This visual representation represents the features learned from successive Convolution Layers within a Neural Network.

Figure 31 & Figure 32 give a visual representation of bus and car at a Level Crossing site respectively. In the first layer, the activation from the Neural Network has retained all the information from the given image (e.g., bus or car). It contains some edges or a collection of apparent edges that is interpretable by human eyes as well. The activations or representations become more complex in deeper layers of the Neural Network. They are more abstract and less interpretable. These complex features correspond to the specific class rather than a visual interpretation of the generic object. The black boxes in the subsequent layers of the Neural Network demonstrate the sparsity of the Network. These black boxes represent that the pattern encoded by the particular filter is not found in this particular location. The Convolution Neural Network is fed with raw data containing specific objects, the subsequent layers within the CNN learn relevant information specific to the class and leaves irrelevant information e.g., the visual appearance of the image. For this reason, any object similar to the given object in the Image regardless of its orientation and size will be predicted accurately. The whole idea of retaining this relevant information but not irrelevant information is inspired by how the human brain works. Humans remember objects in their daily life from generic features related to class but chances are they would ignore very specific details of these objects.

4.1.2 Training using Transfer Learning

There are many different notable models available for Feature Extraction or Transfer Learning e.g., ResNet or Inception etc. The approach is called "Feature Extraction" because it utilises the features when it was trained on millions of images, their respective parameters and values were stored in their "weights".

These weights are used to learn motifs and particular features for the specific new class. As shown in Figure 31 & Figure 32 that only the last few layers are very specific to the class hence the small dataset used along with the pre-trained weights would allow the network to learn more representations using generic features from earlier layers compared with the training from scratch. This approach allows the "freezing of the earlier layers" of the Neural Network and reduces the number of trainable parameters, where trainable layers and parameters are only the last few additional layers added according to the specific application. There is another approach used in Transfer Learning; train the whole Neural Network where every layer is trainable. This may not be an ideal situation where the dataset is comparatively small, still, we shall see how it affects the results compared with the simple Feature Extraction technique.

To choose the most suitable model for a Level Crossing application (or in any other application scenario), it is ideal to train few pre-trained models in parallel and compare which model is most efficient with your dataset. The most accurate model is chosen for further post-processing and its final application at a Level Crossing. For the Level Crossing application, few notable models are selected for their accuracy and state-of-the-art results in computer vision applications. ResNet, Inception, DenseNet, MobileNet, Xception and VGG are chosen for this proposed work and their original weights are available from Keras (Keras, no date). These models would allow a fair comparison for a Level Crossing application and the best model would be chosen for post-processing if required. For fair comparison one particular technique is used for all these mentioned models; the pre-trained weight with no-top is used for initialization of parameters and the last layer is removed to add a Flatten and a Dense Layer at the end of the model according to our required outputs e.g., 4. Image Dataset is the same for each trained model using Transfer Learning techniques. The input shape of an image is 150, 150, 3 for each model as well. The model summary before and after are mentioned along with their results and accuracy graphs.

4.1.2.1 VGG-Net

VGG-Net model has two different versions available as discussed earlier depending on their layers. The VGG-Net along with its pre-trained weights are used for Transfer Learning. Data Augmentations techniques e.g., Rescaling, Shifting or Zoom are used to add diversity in the limited dataset available for model's training. The details of VGG-16 are mentioned in Table 10.

	No. of trainable Parameters (Architecture with no Top)	**No. of trainable Parameters (Architecture with new layers e.g., Flatten Layer and Dense Layer)**	**Accuracy**
VGG-16	14M	8M	Training acc: 74.67% Validation acc: 79.00%

Table 10Details of VGG-16 Network used for Transfer Learning. The trainable parameters and accuracy are mentioned.

The addition of Flatten and Dense Layer increases the number of parameters from 14M to 23M but the trainable parameters are only 8M. These parameters are trained on 40 Epochs (the batch size is the whole dataset and Epoch represent the number of iterations). From one such training, the model achieves an accuracy of 70+% which is significantly high from what models achieve when trained from Scratch. The model has not reached Overfitting as well as is clear from Figure 33.

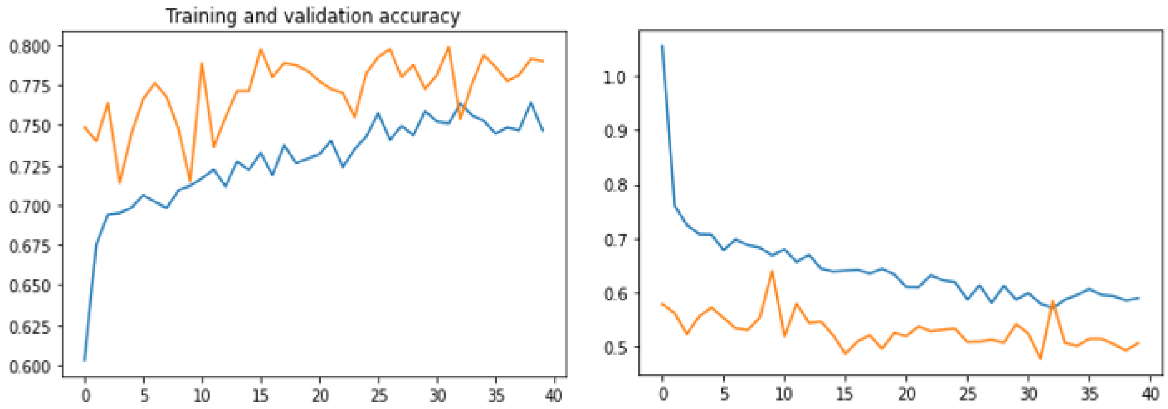


Figure 33Training and Validation Accuracy of the model VGG-16, where blue color represents Training Accuracy and Loss and Orange represents Validation Accuracy and Loss.

Similarly, VGG-19 is trained using the same procedure with the same Dataset and Image Augmentation techniques. The results details are mentioned in Table 11, whereas the Training and Validation accuracy and loss are mentioned in Figure 34.

	No. of trainable Parameters (Architecture with no Top)	**No. of trainable Parameters (Architecture with new layers e.g. Flatten Layer and Dense Layer)**	**Accuracy**
VGG-19	20M	8M	Training acc: 72.83% Validation acc: 76.00%

Table 11Details of VGG-19 Network used for Transfer Learning. The trainable parameters and accuracy are mentioned.

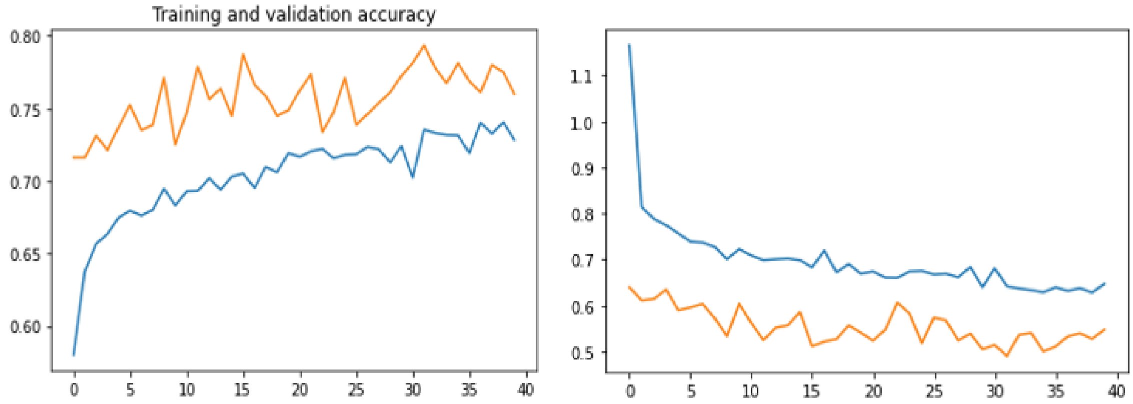


Figure 34Training and Validation Accuracy of the model VGG-19, where blue color represents Training Accuracy and Loss and Orange represents Validation Accuracy and Loss.

4.1.2.2 Other Models

Likewise, the other models are trained using the same procedure and techniques. The top of each model is removed and replaced with a new Dense and Flatten Layer. The weight files are downloaded, which are pre-trained on the original application of the model using millions of images. These weight files are used for the initialisation of the parameters. Image Augmentation techniques on the same dataset used for Image Classification, where some Augmentation techniques involve Zoom or Horizontal Flip. Some selected models are trained using mentioned techniques and the most accurate model is selected for post-processing (if required). The results from such models are mentioned in Table 12 and their visualization is mentioned in Appendix.

	No. of trainable Parameters (Architecture with no Top)	**No. of trainable Parameters (Architecture with new layers e.g., Flatten Layer and Dense Layer)**	**Accuracy Training Acc/ Validation Acc**
DenseNet	7M	16M	79.55% / 87.50%
Inception	21M	38.5M	83.17% / 86.25%
Xception	20.8M	52.4M	80.23% / 85.25%
ResNet	23.5M	52.4M	44.98% / 53.00%
MobileNet	3M	16M	86.02% / 88.00%

Table 12Networks used for Transfer Learning. The number of trainable parameters, training and validation accuracy is mentioned as well.

4.1.2.3 Final Remarks

Different models are used for Transfer Learning to achieve better results and accuracy with a less available dataset. All of these models have their pre-trained weights, which are used for the initialisation of the parameters. These models do not have a top layer rather is replaced with a Dense and Flatten layer. The same Image Dataset containing 8000 images for training and 2000 for validation with 4 categories; Vehicle, Bicycle, Adult and Child pedestrian. These models are trained for 40 epochs and the new weights and history is saved during the training steps. Different models are trained before preferring one particular model for its application or post-processing (if required).

The models achieve accuracy from 44% to 86% using the same procedure and techniques just different model's architecture. The MobileNet achieves the highest accuracy of 86.02%, whereas ResNet achieves the lowest accuracy of about 44.98%. This is precisely the reason why different models should be run before giving preference to one particular model for post-processing or its application. For example, if only the ResNet model, which achieved state-of-the-art results was selected for a Level Crossing application. The model would have achieved an accuracy of about 44% and it would have been a difficult task to further increase the accuracy to even around 70%. Other models achieved an accuracy of more than 70% using the same parameters and number of iterations compared with the ResNet model. If no careful selection of the model has been made at its initial stage, the practitioner would waste lots of computational power and time.

Therefore, the proposed network uses MobileNet for its application at a Level Crossing area. The MobileNet achieved the highest accuracy with less trainable parameters. To further increase the accuracy and avoid the Overfitting problem, the model is retrained where every layer is trainable and no pre-trained weights are used. The idea is to improve accuracy (if possible) and reduce the effect of Overfitting. The new network details are mentioned in Table 12 and the results are visualized in Figure 36.

	No. of Parameters	No. of trainable Parameters	Accuracy
MobileNet	20M	16M	Training acc: 91.90% Validation acc: 88.38%

Table 13Details of MobileNet used for retraining using Transfer Learning technique. The trainable parameters, training and validation accuracy are mentioned as well.

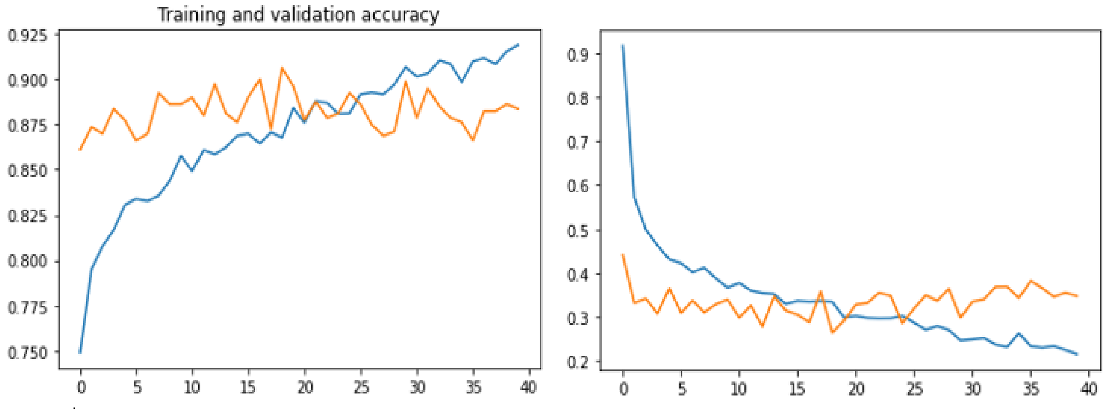


Figure 35Results from MobileNet using Transfer Learning. The training accuracy is about 91% (blue line) and validation accuracy is about 88% (orange line).

The results demonstrate that the accuracy increases from 86% to 91% and Overfitting are reduced as well. The results are achieved when every layer within a model is trainable and no pre-trained weights are used for the initialization of parameters. The model MobileNet is appropriate for Image Classification at a Level Crossing, where the model achieves an accuracy of 90+%. The model is selected for its accuracy and lightweight architecture. MobileNet is fast and can predict images in real-time, which is an essential key feature for a Level Crossing application.

4.2 Object Detection using CCTV

Image Classification is a useful application at a Level Crossing, where images are fed from the Camera. These images are used for predicting the presence of an obstacle at a Level Crossing area. The presence or absence of an object let the circuit decide whether the barrier should remain open or the closing process should be initiated. However, the process of extracting images and feeding them to the Neural Network for prediction is a waste of computation and time given it does not provide more valuable information e.g., location of an object or predicting multiple objects in real-time. The ability of a Neural Network to predict objects in real-time and give precise locations within an image frame is called Object Detection. Object Detection is an upgrade of Image Classification. The dataset for Object Detection is not merely images rather images with labelled bounding boxes around each category. Therefore, each image would contain a corresponding file with its relevant details. Different annotating tools are available to annotate the image or some pre-defined open-source images are available as well. The proposed work contains manually annotated images for each required category e.g., Adult, Child and Bicycle and Vehicle. These bounding boxes allow the Neural Network to learn more precise features and their respective locations along with their labels. To add diversity different augmentations and pre-processing techniques are applied to the dataset as shown in Figure 36. The images in Figure 36 shows different operations e.g., blur, crop, resize and noise are applied to the dataset.

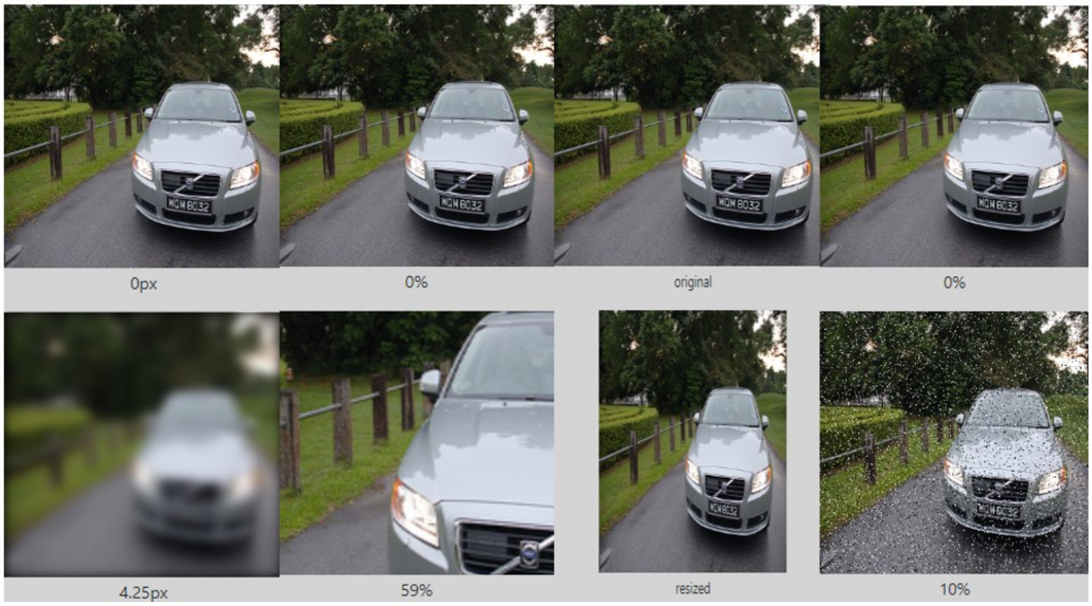


Figure 36Different Augmentation and Pre-processing operation are applied to the dataset. a) Blur of intensity 4.25px b) Crop of 59% c) Resize d) Noise addition of 10%.

The process of training a model e.g., from scratch or using Transfer Learning is similar in principle to Image Classification. For Image Classification, the models from Transfer Learning techniques were more efficient and achieved higher accuracy compared with models trained from scratch. To avoid wastage of time, the Object Detection models are trained using pre-trained models for their application at a Level Crossing. Details of how Object Detection works are discussed in Object Detection.

The preparation of a dataset is an important part of training a model for Object Detection. Different annotation tools are available for labelling and manually defining the bounding boxes around an obstacle, which automatically generates a file required for model training. For the present work, the "LabelImg" tool is used for annotating the images for its application at a Level Crossing. Figure 37 demonstrate a few images from the dataset, which are manually annotated for relevant categories. The manual definition of these bounding boxes generates relevant files used for training the model. The relevant file may have a different file extension e.g., XML or CSV. These files are converted during the training process accordingly.

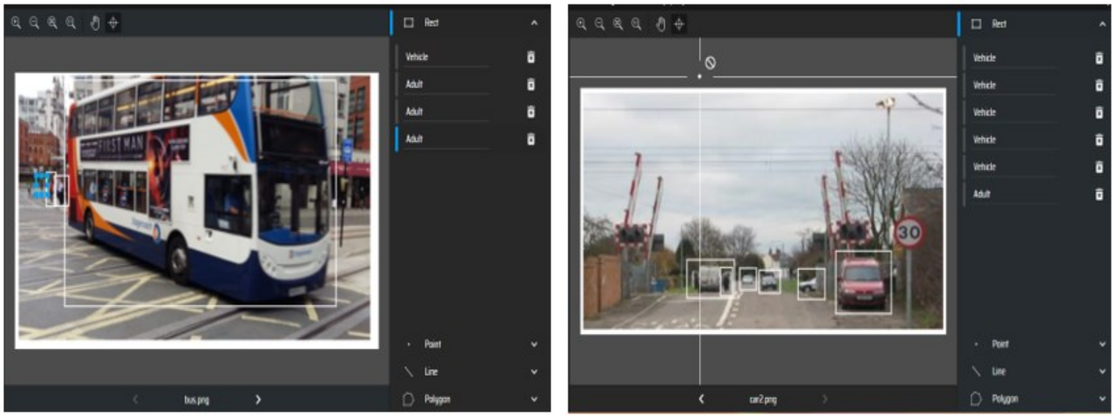


Figure 37Annotated images where relevant categories are defined using bounding boxes. These bounding boxes coordinates are used to automatically generate a file required for training the model.

Using the “LabelImg” tool the dataset for each category is generated; 200 images for each category. These images contain the bounding boxes and their relevant label in the “*XML*” file extensions. The dataset is used to train the models provided by *TensorFlow Model Zoo*, which contain the model architecture and configuration file. These configuration files contain the output number, labels path and path for relevant files required for the transfer learning process e.g., TensorFlow records. The label map and TensorFlow records are the same for the given dataset regardless of what model is used for TensorFlow training. The configuration pipeline is different for each given model and is available when the model is downloaded. These configured files and the dataset are fed with the downloaded model for training. For model evaluation, TensorFlow allows the user to record “*losses*” or use tensor boards to visualize the *loss* during the training process.

4.2.1 Results

For evaluating the performance, the COCO dataset which is a widely used benchmark for Object Detection is used to compare the results for each given model. Below are some models used for training the model for Object Detection. Their respective *Loss* metrics and Average Step time during the training process are mentioned in Table 13. To evaluate and compare the models using transfer learning techniques, the step-size is kept constant at 0.0025 and the value of batch size at 1. The dataset is the same for each model where each category has about 200 labelled bounding boxes. The computation power available for training are

- Ge-Force NVIDIA GPU 1060
- RAM 24GB
- 1 TB HHD and 256GB SSD

The *Loss* value of about two demonstrates the efficiency of the model, where any value less than two makes the model more efficient and suitable for its application for its real-time interference.

Model	Loss (COCO evaluation metrics)	Average Step time (sec)
Efficient-D7	1.15	0.161
RCNN	2.178	0.315
ResNet	0.721	0.133
MobileNet	0.318	0.076

Table 14Brief summary of models used for Object Detection. Their respective Losses and Average Step time are mentioned as well.

Table 14 shows every given model achieves a given benchmark (Loss value of about 2), where RCNN takes the longest time, 0.315 seconds for each step-size with a Loss metric of about 2.178. However, the MobileNet has the lowest *Loss* metric with the shortest Step time of about 0.076 seconds for each step size. As discussed earlier, the MobileNet is lightweight in its application and preferable for real-time application, which is strongly demonstrated in Table 14. Other than *Loss* metrics, the tensor board is used to visualize the results and demonstrate other metrics e.g., Classification Loss, Normalized Loss and Total Loss during its training phase. These results are mentioned in Object Detection using Transfer Learning techniques.

To further improve the accuracy of MobileNet, the Step-size is increased to 50000 steps keeping batch size 1. The results are mentioned in Table 15.

Model	Loss (COCO evaluation metrics)	Average Step time (sec)
MobileNet – 50k Steps	0.092	0.255

Table 15New metrics from MobileNet with an increased step size of 50000.

The average step-time is increased from 0.076 seconds to 0.255 seconds but the *Loss* metrics are significantly decreased from 0.318 to 0.092. The *Loss* metric of about 0.092 is the most efficient model achieved for its application at a Level Crossing. The average time of 0.255 seconds suggests that the model will make multiple predictions in a one-second time frame, which is essential for real-time application. Multiple predictions also suggest that if the system makes a false prediction, it will be superseded with multiple other positive predictions. These multiple predictions will allow the network to re-evaluate its predictions and make a suitable decision before a human can react to the situation.

4.3 Application of RADAR

The Vision System e.g., CCTV camera and its integration with Deep Learning is discussed in detail in **Error! Reference source not found.**. The system can classify, localise or detect specified obstacles at a Level Crossing area with an accuracy of 90+%. The system can work in a dynamic environment and is not affected by the change in texture, orientation or colour. The vision system might have limited performance in a low-light environment, which could be avoided using flashlights around the area.

RADAR is another sensor mostly installed and used as a primary detector at a Level Crossing area within Great Britain. The RADAR system is easy to install and maintain, it is not affected by a change in environmental conditions e.g., rain, snow or light. The long-life expectancy makes it a preferable choice and investment for relevant industries to make RADAR a primary detector at a Level Crossing. Therefore, it is essential to utilise RADAR in the most efficient way along with the Vision System to add another layer of resilience to the new proposed system. These two proposed systems will make an effective “2oo2” interlocking system for its application at a Level Crossing. The limitation of RADAR is a classification between certain objects e.g., it cannot differentiate between harmless cardboard and a high-risk child situation at a Level Crossing. To avoid such limitations, the micro-Doppler signals from RADAR are used and fed to a new CNN, where enough representations are learned to classify specific objects. The micro-Doppler signals are generated from small regular movements from an object, which makes them distinct from other objects.

Traditionally, the RADAR sensor was used to obtain the speed, direction and range of an obstacle which would allow the sensor to post-process the information and roughly classify object categories. Such classification is not accurate neither reliable for its application at a Level Crossing. The classification would often require more than one RADAR sensor to obtain direction and range to make a rough 3D image of an obstacle, which would again allow rough classification of an obstacle and increase the cost of the system. The introduction of the Deep Learning field allows more accurate and reliable classification of RADAR data. Initially, the SAR (Synthetic Aperture RADAR) were proposed and their images were obtained from raw data. These SAR images were fed to the CNN model for classification. These SAR images provide sufficient details and features to learn by the CNN architecture, which is required to classify and localise different objects. However, the SAR images require specific RADAR at a certain height, hence it is not a preferable choice for its applicability at a Level Crossing.

4.3.1 Results

The RADAR sensor is used to generate micro-Doppler signals. The reflected signal from an obstacle produce micro-Doppler signatures and since each object movement differs, their signatures will differ as well. For example, the pedestrian has certain movements from a swing of an arm or legs, while a bicycle has constant movement from the rotation of pedals. Backscattering of signals reflected from objects e.g., bicycle, car and pedestrian is obtained to compute the short-time Fourier Transformation (STFT) of the signal. The STFT of the signal will produce the micro-Doppler signal required for the proposed work. Figure 38 demonstrate three distinct micro-Doppler signals from three distinct objects.

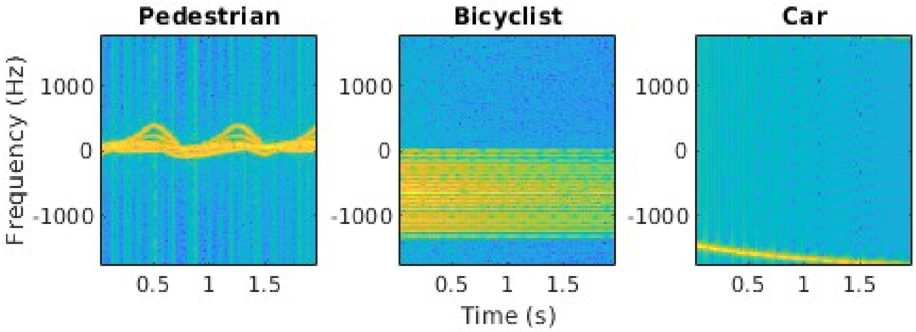


Figure 38Micro-Doppler signals from three distinct object as required for its application at Level Crossing. The Pedestrian generated a signal with most disturbance in a signal while the Car has almost constant signal.

As shown in Figure 38, each signature is unique and distinct from another category. For example, the pedestrian has the strongest movement e.g., swinging arms and legs compared to the pedal movement of bicycle and rigid body of the car. These micro-Doppler signatures provide a dataset for CNN architecture to learn sufficient features and categorise these objects. MATLAB “Phased Array Toolbox” allows the users to generate and simulate the RADAR signals to obtain micro-Doppler signatures and generate a custom dataset. These functions allow the users to define the “Region of Interest” and different varying properties of each object e.g., height and speed. Using the toolbox, the dataset containing 250 signatures for each object for training and 50 signatures for each object for testing is generated for evaluation of the model. The reason for the small dataset is

1. The RADAR has specific objects with varying properties, these properties could only be changed to a certain extent.
2. The MATLAB has memory limitations as well, where the software would not allow more signals to be generated from the RADAR simulated signatures.

The dataset is fed to the CNN architecture with details as mentioned below in Table 16.

	Description	**Accuracy**
CNN for RADAR Classification	5 Convolutional Layer, 5 Max Pooling Layers, 1 Fully Connected Layer	100%

Table 16The first CNN trained on micro-Doppler signatures achieved an accuracy of 100%.

As shown in Table 16, the model is “Overfitting” on the dataset. A more appropriate visualisation of the results is demonstrated in Figure 39.

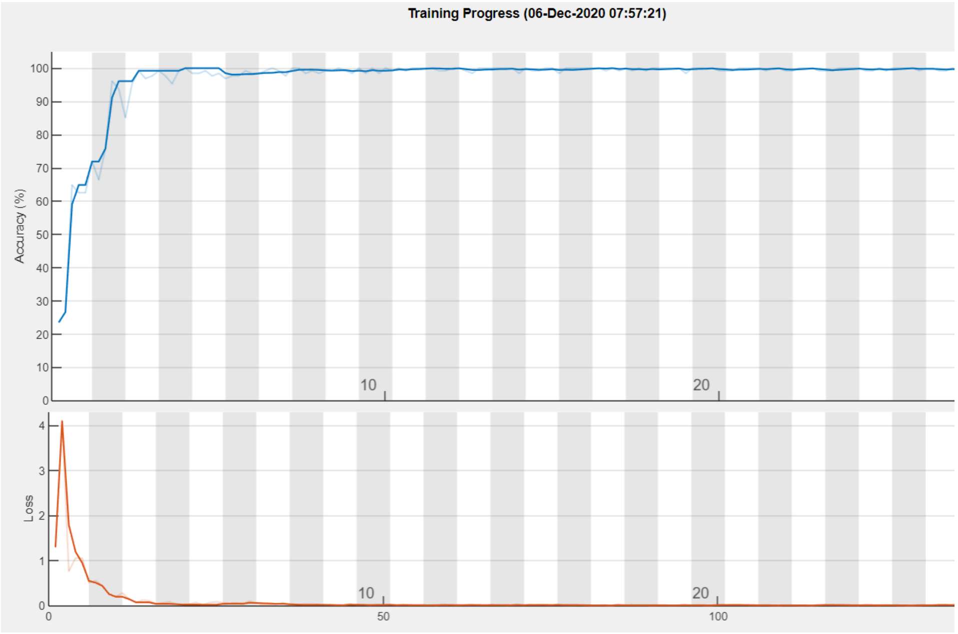


Figure 39Model is fed with 250 signatures for each object to a CNN network with 5 Convolution Layers followed by Pooling Layer. The model is "Overfitting".

As shown in Figure 39, the model stopped learning at the very start of the training process as shown by a constant line. The model did not learn anything after that phase and completely correlated the training data with its labels. To avoid the “Overfitting” problem, the model is fed to a smaller Neural Network. A small CNN architecture is the right choice when the dataset is really small and no other models are available for Transfer Learning. Some other hyperparameters and options for training are changed for trial-and-error before finalising the CNN model with the right accuracy. The new CNN architecture along with its accuracy is mentioned in Table 17.

	Description	**Accuracy**
CNN for RADAR Classification	1 Convolution Layer, 1 Batch Normalisation Layer, 1 ReLU Layer, 1 Max. Pooling Layer 1 Fully Connected Layer, Classification Layer	92%

Table 17Description of new improved CNN architecture proposed to avoid Overfitting. The model achieved an accuracy of 92%.

Figure 40 is used to visualise the results from the given CNN architecture.

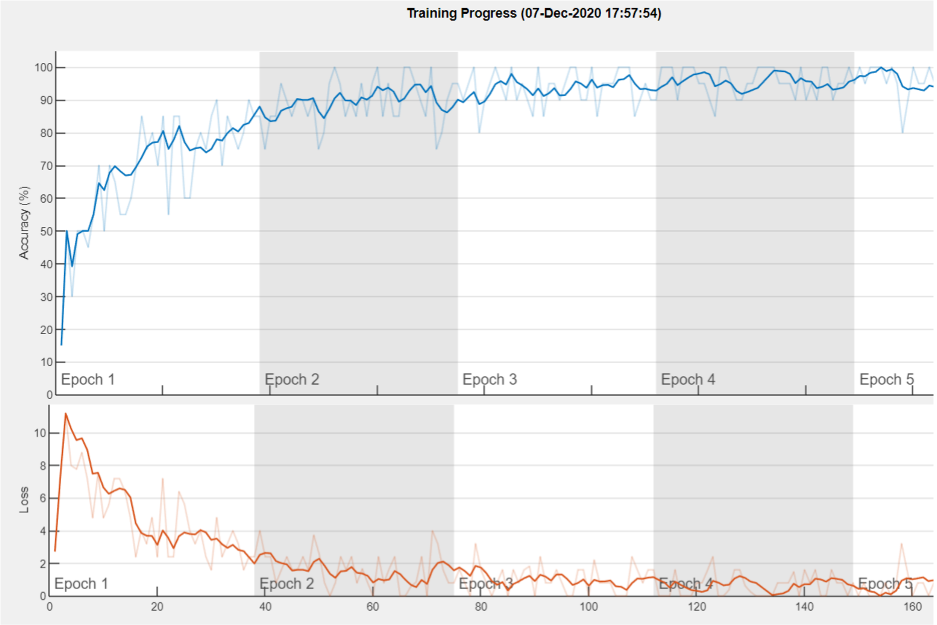


Figure 40A more likely representation of model training using CNN. The model achieved an accuracy of 92%.

Figure 40 shows an actual representation of the model training using a small CNN architecture. The model shows a learning curve throughout the training process, where training accuracy fluctuates to finally achieve an accuracy of about 92%. To evaluate the model, the new dataset called "test-dataset" is used to predict the micro-Doppler signatures. The accuracy on test-dataset achieved an accuracy of 92% as well, which strongly suggest that the model did not "Overfit". To visualize the given results MATLAB generates Confusion Matrix as shown in

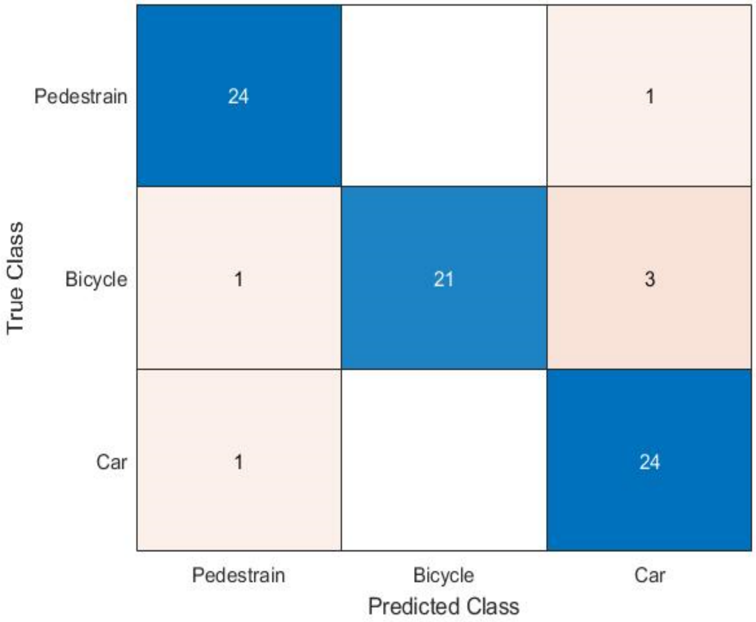


Figure 41Confusion Matrix from the final model, which compares the True and Predicted Class.

The Confusion Matrix shown in Figure 41 represents the True Class and Predicted Class for the given category. The model managed to predict all classes belonging to "Bicycle" but predicted 3 "Car" classes when they belonged to the "Bicycle". Strong evidence for this false prediction is because the micro-Doppler signature from Bicycle is from the movement of pedals and Car is from tire rotation, which is similar in principle. To improve the model's accuracy more data should be fed to the CNN with more variations. Also, the data should be generated using actual RADAR at a physical site, which is from the same distribution rather than using a simulated dataset. However, the model accuracy of 92% strongly suggests its preferability for its application at a Level Crossing.

4.4 OTHER APPLICATIONS

The same "model system" discussed earlier in this work for the efficient operation of a Level Crossing is as well applicable for other applications within Railway Industry. The model is trained for its applicability for both sensors e.g., Vision system and RADAR system. The same trained model can integrate with any Vision system within the Rail Industry. Some key areas within Rail Industry, where the same model can integrate are discussed.

4.4.1 Classification of Passengers at Platform

Often relevant authorities require statistics of specific categories using Railway Platform at a specific time of the service. These statistics should include the passengers with bicycles or passengers with luggage or disabled chair. The data can help relevant authorities to analyse their traffic at a particular Platform and accordingly upgrade the given services or facilities. The statistics will also allow the relevant authorities to analyse what particular Platform is most busy and at what particular hour. All this relevant information is achievable from Image Classifier trained and discussed in Application of CNN- Image Classification. The acquired information is fed to simple yet powerful data visualization software e.g., a tableau that gathers these data and visualises them in relevant graphs providing information more clearly and efficiently.

4.4.2 Tracking any suspicious Behavior

Relevant authorities are always managing the traffic at the Platform or within the train to detect and respond to suspicious behaviour e.g., suicidal attempts or violence. Busy traffic at the Platform often makes it impossible for a manual operator to detect such activity in time or respond to it efficiently. To detect such occurrences based on certain instances e.g., random movement of a person near rail lines or people gathering at one place. These instances are easily trackable using the Deep Learning model, which tracks any random or unwanted movements in the pre-defined area. The threshold value is used to alarm the system if the detection is stronger than the given threshold value. This will give enough time to manual operators at the Platform to respond and avoid any unwanted situations. The model can track such objects at Platform, which makes it easy for relevant authorities to respond in a short timeframe.

These are some applications, which are achievable using the same Deep Learning model. Using different pre-trained models available along with their relevant weight files are used to train models for a given application at a Level Crossing. Other applications may include a census of passengers at Platform, managing passengers at Platform at the time of train arrival or tracking objects at a private accessed site at Platform etc.

5 Safety Risk and New Interlocked System

The proposed system does not demand the installation of new sensors at a Level Crossing rather it integrates with the existing sensing system. Hence, the Risk associated with the sensors e.g., CCTV and RADAR for its installation is not discussed in this work. The proposed system integrates Deep Learning technology, which is trained at a different site. Once trained, the system is deployed at a Level Crossing site using the existing computers at the site. The computation power available from the existing computers at a Level Crossing is sufficient to process the video loop from CCTV for its operational cycle. Likewise, the RADAR system requires some post-processing before it could be used for Deep Learning models as discussed earlier, which is achievable from the given computational resources.

5.1 Data Collection

The problem with Risk assessment is the collection of data, which is required for initial Risk Assessment, Accurate Quantitative Assessment and is often used to describe variation in Qualitative Assessment. Therefore, the authorities try to gather data quarterly for Risk Assessment and the gathered data allows them to assess, review and monitor their systems and update or upgrade accordingly. This is an expensive method of data collection and requires expertise. The proposed method to classify and detect obstacles not only automate the operational cycle at a Level Crossing but gathers these predictions and evaluations as well. The Deep Learning model allows the user to save its predicted values, bounding boxes and statistics over a defined time frame. The numerical data accuracy depends on the accuracy of the network. This predicted data is valuable to the authorities for their Risk Assessment and other statistical purposes. The gathered data could be used with other software such as Tableau for better visualisation and representation. Such representation will allow authorities to get a census of each category particular to each type of Level Crossing to analyse what particular category frequently uses or misuses a Level

Crossing. Accurate data are essential for quantitative assessment, which is a preferable and reliable method of Risk Assessment, where authorities can successfully perform a quantitative assessment. This will allow authorities to assess and review their Risk Assessment for better and safer implementation of the systems and provide new updated and relevant data for re-training the network if required. The model performance is demonstrated in the acquired data if the system cannot classify or detect a particular type of obstacle it will be reflected as well. This will allow the authorities to gather data from the same distribution e.g. Level Crossing particular to the objects which are using the Level Crossing and re-train the network for more accurate results. This update will give more accurate numerical statistics for quantitative risk assessment. Therefore, the proposed system can adapt and re-train itself for new changes with time but also provide accurate data for Risk Assessment at no extra cost and expertise.

As discussed earlier, the two main categories of Risk Assessments are General and Specific. For General Assessment qualitative method is used to evaluate the Risk associated with the Level Crossing and for Specific Assessment, FMECA is used to perform Risk Assessment.

5.2 General Risk Assessment

The fundamental questions that should be addressed to perform Risk Analysis.

1. What can happen and Why?

Some possible risks are identified and discussed below

- Power Breakdown: The system is at risk of shutting down due to some power breakdown or in case of failure.
- System Breakdown: The system available may have some unwanted glitches and pause in the simulation time.
- Vandalism: Level Crossing users, environment or wildlife (if any) may cause damage to the system.
- Sensors: Since the proposed system relies on the input from the existing sensors, any failure or fault in the sensor will eventually affect the functionality of the system.
- Overfitting: The Deep Learning model may get affected from “overfitting”, where a model learns perfect representation from “training” data only and does not work on new data or “scenario” in case of a Level Crossing scene.

2. What are the failure effects?
   - Power Breakdown: The system stops and sensors are unable to detect or scan the Area of Detection, which fails the system as a whole. The effect may be “catastrophic” if it happens.
   - System Breakdown: The system breakdown or glitches may lead to “no” or “slow-response” time. This will affect the real-time detection as required by the Level Crossing application. The effect of such a risk is “critical” to the system.
   - Vandalism: The users may steal or wildlife may destroy the system, which will leave a Level Crossing Area with no effective system to scan, detect or perform an operational cycle at a Level Crossing. The effect caused by vandalism is “marginal”.
   - Sensors: The risk associated with Sensors, if not assessed properly may cause a fault or failure of sensors. Such failures will disrupt or eventually stop the operational cycle at a Level Crossing. The risk level is dependent on the risk caused by the sensors.

- Overfitting: If the system is affected by “overfitting”, the system will likely misclassify obstacles at a Level Crossing during its operational cycle. The risk associated with such a system is “negligible”.

3. How likely is it to happen?
    - Power Breakdown: The frequency of such a system failure within the UK is “improbable”.
    - System Breakdown: The frequency of system breakdown or small glitches are “occasional”.
    - Vandalism: The risk caused by vandalism is either “probable” or almost “improbable” depending on the location of the Level Crossing.
    - Sensors: The frequency of sensor failure or fault is “occasional”. Other associated risks are dependent on the risk level of sensors themselves according to their manufacturing standards.
    - Overfitting: The risk caused by the issue of “overfitting” to a Level Crossing application is near to “improbable”.
4. What is the level of risk?
    - Power Breakdown: Even though the risk associated with Power Breakdown is “catastrophic”, because if power breaks, the whole operational cycle stops and no scanning is possible neither is the movement of barriers. For such occasions, the recovery time might be slow and during such time it is “probable” that many accidents may occur. However, the Level Crossing site has emergency generators at the site for such occasions. Hence, the risk caused by Power breakdown is “improbable”.
    - System Breakdown: The small glitches or occasionally slow computation may affect the functionality of a Level Crossing. However, the computation power to process 15 fps is enough for the system to operate effectively. Any false or no detection does not affect the operational cycle and false prediction will be replaced with another new prediction without a human or train driver to notice since the human reaction time is about 2-4 seconds. To completely overcome the occasional glitches, the system available at a Level Crossing could be upgraded to GPUs.
    - Vandalism: Protecting the system from wildlife and some users who may steal or destroy the sensors is essential to ensure the continuous functionality of the system. The Level Crossing sites have existing protective housing for CCTV and an enclosed box for computers in a corner. This ensures the safety of such systems from vandalism, hence the risk associated with vandalism is negligible.
    - Sensors: The occasional damage caused by sensors is avoided by considering the risk assessment of each sensor individually and ensuring each risk is monitored and reviewed in the defined time frame. Also, the primary sensor (RADAR) is supported by a secondary sensor (CCTV), which will add another layer of resilience to the system. In case of failure or fault in one sensor, the other sensor can act as the primary detector during the recovery time. The present work uses the “2oo2” approach for the Interlocking system to ensure a safer and reliable system is installed for its application at a Level Crossing. A brief discussion on the “2oo2” Interlocking system is discussed below.
    - Overfitting: The proposed system can learn and re-learn from the given data to ensure the robustness of the system. If the given system is unable to detect certain situations, new data for these particular situations is fed to the network to retrain to detect and classify

accordingly. In the meantime, the false detection will be superseded by many updated predictions of the obstacle. Hence, the risk associated with overfitting is negligible.

5.3 Specific Risk Assessment

For specific Risk Assessment, the FMECA technique is used for its flexibility in Risk Assessment stages; Identification, Analysis and Evaluation. As mentioned earlier, the FMECA should address six fundamental questions for each possible failure mode. Below is the discussion of every possible failure mode during the training, processing and the system itself.

In the design process, the system may have several issues that need to be addressed and resolved. The data used for training may not be sufficient or may not be from the same distribution e.g. Level Crossing. Same distribution data is required to avoid any bias in the network, likewise, enough data should be available to learn sufficient representation for classification and detection. To avoid any bias and underfitting, two different techniques are utilized for this work. Firstly, two different datasets are downloaded from ImageNet which are combined to form one large dataset used for the Image Classification model trained from scratch. This will provide sufficient data to learn representation. Secondly, the transfer learning techniques are used where pre-trained models (trained on millions of images) are used for their application at a Level Crossing. This will allow the Neural Network to retain the abstract features learned from those millions of images and utilize them to create higher motifs and groups of motifs for a particular category. For the RADAR dataset, the MATLAB functions are used to simulate the RADAR signals, which are used to generate micro-Doppler signals. These micro-Doppler signals are specific for each distinct category and are used to train a Convolutional Neural Network for classifying RADAR signals.

Creating a Neural Network from scratch is an art and requires time and careful engineering to design the architecture. The poorly designed architecture will fail to classify objects accurately and will unnecessarily create deep networks without increasing efficiency. Firstly, CNN is designed from scratch and hyperparameters are optimized to further increase the accuracy of the network. The model with an accuracy of about 75% is sufficient to demonstrate its effectiveness in its application at a Level Crossing. Afterwards, the architecture of a pre-trained Neural Network is used, which have been trained and evaluated on different applications and a dataset containing millions of images. A similar type of Neural Network will be used for training on the RADAR dataset. The new proposed Convolutional Neural Network is still in its early stage, where it is not widely used and no large dataset is available, therefore, the proposed work generated a small amount of dataset to train the CNN and train a model. The dataset is open-sourced for other interested researchers to train and improve the proposed model.

The training of this Neural Network requires high computational power for efficient training. The high computational power is achieved from GPUs, hence risk associated with hardware is considered as well during the processing phase of the system. The computer system must have an updated Risk Assessment for all its wiring and environment. Any failure within the system will affect the training and ultimately the whole system. The Risk Assessment of these systems is updated and monitored to ensure no failure modes are expected during the training and optimizing of the Neural Network.

Once trained and tested, the final Neural Network file is used for interference at a Level Crossing site. These systems are integrated with the sensors, where a continuous stream of data from CCTV and RADAR is fed to the Neural Network to perform classification and detection. These classification and detections are used to automate the operational cycle of the Level Crossing and also collect the data for further Risk Assessment. To ensure the system works properly, relevant Risk Assessments are performed of the whole system as well. For example, the system should be regularly monitored and updated and it should be installed in a safe component, where it is not accessible by the public. It should be protected from vandalism as well. The sensors should be working properly along with the barriers and with any electrical component used to automate the operational cycle. Before implementation at a real site, the system should be tested at a remote physical site to ensure the safety and reliability of the system. Several test scenarios should be considered which should replicate the Level Crossing site to test and evaluate the system in a complex environment. These test plans and techniques are used to ensure the whole system is working properly before its final implementation at a Level Crossing site.

Table 18 is used to summarize the Risk associated with the application of Deep Learning at Level Crossing. Note: The Specific Risk Assessment is a descriptive analysis of the Risk associated, hence, the present Table 18 only uses the General Risk Assessment for its summary.

Risk	Failure	Effect	Probability	Level
Power Breakdown	Shutting Down of System	Sensor and Operational Cycle	Improbable	Catastrophic
System Breakdown	Unwanted Glitches or Pause	Response Time	Occasional	Critical
Vandalism	Damage to System	Whole System	Improbable	Marginal
Sensor	System Failure	Sensor and Operational Cycle	Occasional	Catastrophic
Overfitting	Ineffective System	False Positives	Improbable	Negligible

Table 18A summary of General Risk Assessment for Deep Learning application at Level Crossing.

5.4 INTERLOCKING SYSTEM

The final proposed system uses the “2oo2” approach for the Interlocking system at a Level Crossing. The Interlocking system uses Safety Integrity Level- SIL, which is a numeric approach to describe the safety and reliability of the system. SIL considers the failure severity and frequency during the operational lifecycle to ensure the right SIL level is assigned to it. For example, a subsystem whose percentage of failure is above 60%, can be assigned a SIL-2 or SIL-3 level depending on the number of hazardous faults in the subsystem. A subsystem of SIL-2 combined with another subsystem of SIL-2 or SIL-3 can secure a SIL-4 level for the whole proposed system, which ensures that the system is fail-safe. The proposed work has two subsystems e.g., CCTV and RADAR with both SIL-2 since they have a high lifecycle with rare chances of faults in the system. The input of these two subsystems is fed to two individual software subsystems, which are fed with the outputs of previous subsystems. The software subsystem has a SIL-3 level since they are trained and tested with an accuracy of more than 60% and “occasional” chances of failure or faults. The whole system contains two pairs of subsystems e.g., pair of sensors and a pair of software subsystems. An illustration of the proposed system is given here in Figure 42.

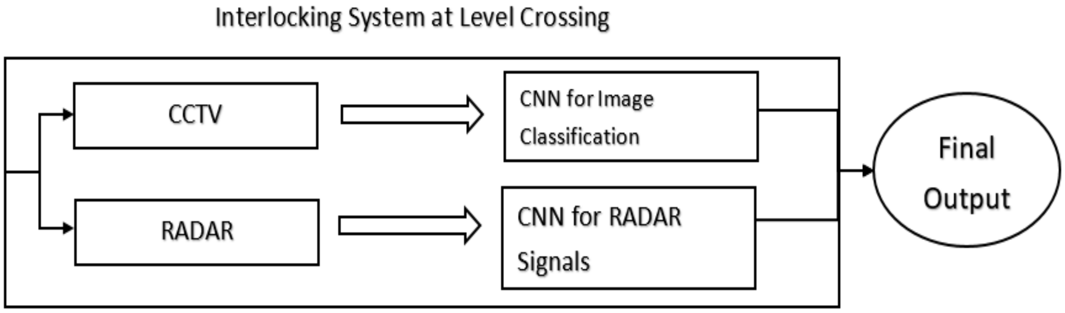

Interlocking System at Level Crossing
CCTV
RADAR
CNN for Image Classification
CNN for RADAR Signals
Final Output

Figure 42Proposed Interlocking system, which use "2oo2" approach to provide SIL of level 4.

The proposed system shown in Figure 42 has two subsystems; sensor subsystem and software subsystems. The sensor subsystems have SIL-2 to SIL-3 rating and the software subsystem has SIL-3 level, which means the Interlocking system has a SIL-4 rating. SIL-4 level ensures that the system is safe and reliable with fail-safe mechanisms. The two outputs are not dependent on each other, rather both sensors provide different information, which is processed by different software to predict one output. The real-time detection where CNN is processing at a 15fps rate ensures that multiple predictions are processed every second and any false positives (if any) will be replaced with another prediction before alarming the train driver or disrupting the operational cycle at a Level Crossing.

6 CONCLUSION & FUTURE WORK

The final chapter of the thesis contains a summary of all contributions discussed earlier in this work. Certain limitations and improvements are discussed along with any proposed future work.

6.1 INTRODUCTION

The Railway Industry, which is still a primary choice of transport for many people, demands expansion, continuous improvement and upgrading of the existing system. The expansion and upgrading of existing technologies are to tackle the systems capacity issue. Railway Industries are managing very large-scale projects and initiated many new ones as well e.g., Control Period 6 (CP-6). These projects are responsible for Rail lines expansion and generating many new platforms within towns and cities. Similar expansion projects are initiated by European authorities, where relevant authorities predict that Rail lines particularly High Speed (category I and II) rails will double in length by 2020 compared with its length in 2008. Such an expansion within the Rail Network will certainly increase the number of junctions and Level Crossings. Level Crossings are often associated with high risk and many accidents and fatalities are recorded at such sites. In 2016/17 the authorities recorded 6 fatalities, 400 near-misses and about 77 reported incidents of shock and trauma. Authorities have suggested that most of these incidents are from misuse of Level Crossings and the current state of art safety devices do not provide adequate coverage on High-Speed lines. The UK risk mitigation policy is to ensure no Level Crossings are installed on High-Speed lines, and wherever possible, the existing Level Crossings should be removed. Such an approach is not feasible with the given landscape of the UK, where the need for a Level Crossing is inevitable. Alternatively, the authorities suggested that some new innovative technology should be proposed, which can automate the process and reduce the misuse of a Level Crossing.

To avoid misuse and automate the operational cycle of a Level Crossing, the proposed work integrates Deep Learning technology with the existing system. To prefer one particular choice of sensor, the proposed work gives a detailed discussion of primary choices of sensing systems and other potential sensors for Obstacle Detection at a Level Crossing. Every sensor has certain limitations, for example, most of these sensors are unable to detect objects particularly small objects e.g., a child. The proposed technology e.g., Deep Learning integrated with a Vision System can learn representations from the given labelled data. After training the model, the system can classify, detect and localize objects regardless of their orientation, size, position and scale. The model for Image Classification achieves an accuracy of 91.90% with the MobileNet model trained using Transfer Learning techniques. For Object Detection, different models were trained and loss metrics were used to evaluate their performances. The MobileNet model achieved loss metrics of 0.092. The CNN model used to classify RADAR signals achieves an accuracy of about 91%. The model's accuracy strongly suggests the capability of the system to detect and classify objects within a Level Crossing area. However, successful risk reduction still relies on effective communication between the signaler (or system) and the train operator. Traditional methods such as AWS magnet to alert the driver or even TPWS to automatically brakes the train will not be cost-effective. The ETCS system currently deployed within Europe is a preferable choice since all three levels of implementation provides enhanced protection and effective continuous communication using GSM-R radio to train driver cap when compared with standard AWS and TPWS systems within the UK.

The present work discusses the Risk associated with the installation and maintenance of sensing systems using the RAMS MANAGEMENT system. The General and Specific Risks are discussed which covers the sensors installation, maintenance and certain Risks associated with the Deep Learning technology. The present work discusses the “2oo2” approach used for the Interlocking system at a Level Crossing. Two subsystems with SIL2-SIL3 are combined to propose a system with SIL-4 level, which ensures that the system is safer and reliable with fail-safe mechanisms. Finally, the present work also discusses certain other applications using the same Neural Network trained on CCTV images. The models trained using Deep Learning are applied in other aspects of the Rail Industry e.g., census and classification of passengers at platform etc.

6.2 Research Findings

Many objectives were set at the start of the thesis, these objectives were externally set for the given project. This section will explain why these objectives were set and how they are addressed.

1. *Determine the most appropriate sensing system for a* Level Crossing *Application*

Section 2.3.1 Obstacle Detection System discusses potential sensors for their applicability at a Level Crossing along with the functionality, suitability and limitations for its application at a Level Crossing. It mentions traditional sensors e.g., Inductive Loops, which are installed inside rail lines. Such sensors make installation and maintenance very expensive. It further discusses non-intrusive sensors which are installed outside rail lines and does not disrupt rail networks during installation and maintenance. Among these non-intrusive sensors, the Vision System e.g., CCTV and RADAR systems are the primary choice of installation within Great Britain. The proposed work utilises these two mentioned sensors because of their suitability, low maintenance and installation cost. The CCTV and RADAR sensors are already installed at most sites of Level Crossings within Great Britain, which will significantly reduce the installation cost for the implementation of the proposed system.

2. *Development of a two-layer sensing system*

A robust approach for the sensing system at a Level Crossing is to develop a two-layer sensing system, where one fails the other can still be operational. The Vision system can classify, detect and localize obstacles e.g., Pedestrian and Vehicles at a Level Crossing area. The functionality of CCTV is reduced in low-light and heavy weather conditions, which could be avoided with a proper lighting box attached to the CCTV sensor. The Deep Learning technology integrated with the Vision System can successfully classify between a small cardboard box and a child because it depends on the representations learned rather than pixel values compared with background pixels using traditional algorithms. The proposed work uses micro-Doppler signatures for Convolutional Neural Network to classify signals from RADAR sensors as well. Traditionally, the RADAR was used to measure the speed, direction and range of an object and these primary measurements were used to classify objects at a Level Crossing. However, the new proposed system learns distinct representations from RADAR signatures using micro-Doppler signatures. These two subsystems are used for the "2oo2" approach of the Interlocking system at a Level Crossing, where both subsystems are processing two different sets of information using two different models to give one output for one particular application.

3. *Determine the most appropriate algorithm for detection and classification of obstacles at a* Level Crossing

A detailed survey of all potential algorithms for post-processing the signals are discussed in 2.3.4 Obstacle Detection Algorithms. Traditional algorithms classify objects based on their pixel values, a change in the background pixel and the new pixel value represents the presence of an object. These sensors cannot adapt to the strong dynamic backgrounds, backgrounds with vivid textures and cannot classify between a child and harmless small cardboard object. Therefore, the proposed work introduces Deep Learning technology for its application at a Level Crossing. Deep Learning technology is integrated with both the Vision system and RADAR. It can be trained on models designed from scratch or using the Transfer Learning technique. Transfer Learning technique uses a pre-trained model, which is previously trained on millions of images and achieved state-of-the-art results in their respective competitions. Results from these different techniques demonstrate what particular method is preferable for its application at a Level Crossing. MobileNet trained using Transfer Learning techniques achieved an accuracy of 91%. The Object Detector uses MobileNet as well and achieved a loss metric of 0.0092, whereas, the CNN model trained on RADAR signals achieved an accuracy of 91%. These metrics indicate its effectiveness and reliability for its application at a Level Crossing.

4. *Ensure that the applied system is applicable both in theory and practice*

The dataset used to train models for Image Classification and Object Detection is further divided into training and validation sets. The training set is used to train and learn representations from the labelled data, which is tested on a validation dataset. If the model cannot predict correctly on the validation set but achieves higher accuracy on a training set, it means that model is Overfitting. Data augmentation techniques are introduced to further add diversity and avoid any bias. Such techniques improve the accuracy of the system without the risk of Overfitting. Once trained and validated, the model is tested with a new dataset called the test dataset. The test dataset is an actual representation of the Level Crossing area with different objects present. These images and videos are taken from the local Level Crossing area where the project was ongoing. The results from the test dataset represent how well the model works at the site of deployment. The results achieved from the model as mentioned earlier demonstrates the effectiveness of the model for its deployment at the site.

5. *Ensure the applied sensing system and algorithm outperforms the traditional approaches.*

The traditional sensing system and algorithms have certain limitations as discussed earlier in their relevant sections. Sensing systems mostly disrupt the railway network during its installation and maintenance, which is a wastage of time and money. Non-intrusive sensors are costly and are unable to classify small objects e.g., a child and a harmless cardboard box. These sensors require some post-processing techniques to classify objects most effectively. Traditional algorithms are used to compare pixels with background pixels to determine the presence of an object. Such approaches are not efficient for the dynamic background or background with strong textures. Integration of Deep Learning technology with chosen sensors e.g., Vision System and RADAR learns representations from the given labelled data and does not directly depend on pixel values. The representations are a strong demonstration of objects regardless of their size, position and orientation. Once trained, the model can be deployed at a Level Crossing site. The model is integrated with traditional sensors; therefore, the installation cost is significantly reduced. The same model could be deployed and used for different applications e.g., census of passengers and classifications of platform users.

6.3 Future Work

The section highlights and discusses future work in this particular area of research.

6.3.1 Ensemble of Neural Networks

Two different Neural Networks are trained, where one is trained on Images from CCTV and the other on micro-Doppler signals from RADAR. These two models are trained and deployed separately with their respective data types; however, one new ensemble model could be trained where the model is trained on both types of data and accordingly make decisions for its application at a Level Crossing. The Ensemble model will use both outputs to give one particular output accordingly.

6.3.2 DATASET improvement and expansion

To improve model accuracy, a more diverse dataset is required, where the new dataset should be from the same distribution. Relevant authorities should be briefed about the present work and its potential, where they can cooperate for data collection. A new dataset should be gathered from CCTV video present at a Level Crossing site, the new dataset will precisely represent the real-world and avoid any prejudice and bias for its application at a Level Crossing.

For the RADAR sensor, the relevant authorities should generate micro-Doppler signals using relevant RADAR sensors at a Level Crossing site compared with the simulated dataset. The generated dataset will represent real-world scenarios more precisely with more diversity at a Level Crossing site, which will achieve more accuracy.

6.3.3 Model’s Training

The field of AI is emerging with new techniques and state-of-the-art results. Future research should focus on such models and analyze if the new given model can achieve higher accuracy at low computation cost within a short time frame. Many researchers are now focused on techniques or mathematical formulation where the new model can achieve more accurate results with a small dataset and shallow Neural Network with low computation and training time. Such models should be trained for Level Crossing applications and replaced if necessary.

6.3.4 Effective Communication System

Standard AWS and TPWS systems within the UK do not provide an effective communication method between track and train. The TPWS is not cost-effective as well since it should be installed at every Level Crossing if chosen for its applicability. The present ETCS system currently deployed within Europe is an effective communication method, where GSM-R radio is used for continuous communication with the train driver. Such methods and their specific applicability are an interesting area of research and future work for the present proposed system.

6.3.5 One-Shot Model For RADAR

The RADAR mentioned in this work has no available dataset online, which could be used to train large Neural Networks. Researches should focus on its application using a small dataset. One-Shot Model is specifically designed to work with the model where the availability of the dataset is limited. The model's accuracy will help relevant authorities to deploy the model and during its deployment phase collect data for training a larger network if required. The dataset used in this work is made public for other researchers to continue the work.

6.4 Concluding Remarks

The present work in this thesis contributes to the safety system at Level Crossings within Great Britain. The present work discusses traditional sensors and their respective algorithms along with their limitations for their applicability at a Level Crossing. RADAR and Vision System, which is already a primary choice for many relevant authorities in Great Britain, is preferred for its applicability because of low maintenance and installation cost. The post-processing techniques have certain limitations because they rely directly on pixel values, which is not preferable for situations with a dynamic background. Deep Learning technology that learns representations from given labelled data is preferred and adopted in this work. Integration of Deep Learning technology with Vision System and RADAR achieves higher accuracy and demonstrate an effective model for deployment at a Level Crossing. The present work also discusses the Risk associated with the given sensors and algorithms. It mentions and discusses the associated Risk using the RAMS Management system. Finally, the thesis concludes with the potential future work and other applications within the Rail Industry, which is possible with the trained model.

7 References

A. Klein, L., K.Mills, M. and R.P. Gibson, D. (2006) *Traffic Detector Handbook*. 3rd edn. Available at: https://www.fhwa.dot.gov/publications/research/operations/its/06108/06108.pdf.

Addabbo, T. *et al.* (2016) 'Reliability and Safety Considerations Affecting the Design of a Radar Based Railway Crossing Level Passage Monitoring System', *14th IMEKO TC10 Workshop on Technical Diagnostics 2016: New Perspectives in Measurements, Tools and Techniques for Systems Reliability, Maintainability and Safety*, pp. 290–293. doi: 10.21014/acta_imeko.v5i4.419.

Allili, M. S., Bouguila, N. and Ziou, D. (2007) 'Finite Generalized Gaussian Mixture Modeling and Applications to Image and Video Foreground Segmentation', in *In Fourth Canadian Conference on*

Computer and Robot Vision (CRV'07). IEEE, pp. 183–190.

Amaral, V. *et al.* (2016) 'Laser-Based Obstacle Detection at Railway Level Crossings', *Journal of Sensors*. doi: 10.1155/2016/1719230.

Amir (2015) *How does pre-training improve classification in neural networks?*, *StackOverflow*. Available at: https://stackoverflow.com/questions/34514687/how-does-pre-training-improve-classification-in-neural-networks (Accessed: 13 May 2020).

An, M., Baker, C. and Zeng., J. (2005) 'A fuzzy-logic-based approach to qualitative risk modeling in the construction process', *World journal of engineering*, 2(1), pp. 1–12.

An, M., Lin, W. and Stirling, A. (2006) 'Fuzzy-reasoning-based approach to qualitative railway risk assessment', *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of rail and rapid transit*, 220(2), pp. 153–167.

Ann, L. M., Rausand, M. and Utne, I. B. (2009) 'Integrating RAMS engineering and management with the safety life cycle of IEC 61508', *Reliability Engineering & System Safety*, 94(12), pp. 1894–1903.

Apheby (2016) *TPWS*, *Network Rail*. Available at: https://safety.networkrail.co.uk/jargon-buster/tpws/ (Accessed: 2 October 2020).

Avizienis, A. J., Laprie, C. and Randell, B. (2001) *Fundamental Concepts of Dependability*.

Baldi, P. and Sadowski, P. (2013) 'Understanding Dropout', (1), pp. 1–9.

Becker, D. (2018) *Rectified Linear Units (ReLU) in Deep Learning*, *Kaggle*. Available at: https://www.kaggle.com/dansbecker/rectified-linear-units-relu-in-deep-learning (Accessed: 30 April 2020).

Birtles (2017) *Design Scope Frieght Train Lengthening Southampton*.

Boureau, Y., Ponce, J. and LeCun, Y. (2009) 'A Theoretical Analysis of Feature Pooling in Visual Recognition'.

Bousquet, O. and Bottou, L. (2007) 'The Tradeoffs of Large Scale Learning', *Advances in neural information processing systems*, 20, pp. 161–168.

Bouwmans, T. (2009) 'Subspace Learning for Background Modeling: A Survey', *A survey. Recent Patents on Computer Science*, 2(3), pp. 223–234. Available at: https://www.ingentaconnect.com/content/ben/cseng/2009/00000002/00000003/art00005.

Bouwmans, T. (2011) 'Recent Advanced Statistical Background Modeling for Foreground Detection - A Systematic Survey', *Recent Patents on Computer Science*, 4(3), pp. 147–176.

Brosch, T. and Tam, R. (2015) 'Efficient Training of Convolutional Deep Belief Networks in the Frequency Domain for Application to High-Resolution 2D and 3D Images', *Neural Computation*, 27(1), pp. 211–227. doi: 10.1162/NECO.

Bucak, S. S., Gunsel, B. and Gursoy, O. (2007) 'Incremental Non-negative Matrix Factorization for Dynamic Background Modelling', in *Proceedings of the 7th International Workshop on Pattern Recognition in Information Systems*, pp. 107–116. Available at: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0002425501070116.

Cao, Z. *et al.* (2017) 'Realtime multi-person 2d pose estimation using part affinity fields.', in *In*

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299. doi: 10.12693/APhysPolA.106.709.

Chen, C. *et al.* (2015) 'DeepDriving : Learning Affordance for Direct Perception in Autonomous Driving', *ICCV*.

Chen, V. C. (2011) *The Micro-Doppler Effect in Radar–Norwood*. Available at: http://aerosociety.com/Assets/Docs/NAL/Book Reviews/AeroJournal_Feb2012.pdf.

Chen, Y. (2012) *Improving Railway Safety Risk Assessment Study*.

Chinmayi, K. A. *et al.* (2017) 'A Technical Review on Background Subtraction and Object Tracking on the Detection of Objects', *International Journal of Innovative Research in Computer and Communication Engineering*, 5(5), pp. 65–67.

Chollet, F. (2017) 'Xception: Deep learning with depthwise separable convolutions', *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.

Choromanska, A., Henaff, M. and Mathieu, M. (2015) 'The Loss Surfaces of Multilayer Networks', 38.

Cires, D. and Meier, U. (2012) 'Multi-column Deep Neural Networks for Image Classification', *Proceedings of the CVPR*.

Colangelo, P. *et al.* (2018) 'Exploration of Low Numeric Precision Deep Learning Inference Using Intel® FPGAs', in *In 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, pp. 73–80. Available at: http://dl.acm.org/citation.cfm?doid=3174243.3174999.

Cortes, C. and Vapnik, V. (1995) 'Support Vector Machine', *Machine Learning*, 20(3), pp. 273–297.

Cribbens, A. H. (1987) 'Solid-state interlocking (SSI): an integrated electronic signalling system for mainline railways', *IEE Proceedings B - Electric Power Applications*, 134(3), pp. 148–158. doi: 10.1049/ip-b.1987.0024.

Dai, J. *et al.* (2016) 'R-FCN: Object detection via region-based fully convolutional networks', *Advances in neural information processing systems*, pp. 379–387. Available at: http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf.

Dalal, N. and Triggs, B. (2010) 'Histograms of Oriented Gradients for Human Detection', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, pp. 886–893.

Darlington, P. (2017) *Obstacle detection for level crossings*, *Rail News*. Available at: https://www.railengineer.co.uk/2017/05/30/obstacle-detection-for-level-crossings/ (Accessed: 26 February 2020).

Dauphin, Y. N. *et al.* (2014) 'Identifying and attacking the saddle point problem in high-dimensional non-convex optimization', pp. 1–9.

Deng, L. (2014) 'A tutorial survey of architectures , algorithms , and applications for deep learning', *APSIPA Trans. Signal Inf. Process*, 3(2), pp. 1–29. doi: 10.1017/ATSIP.2013.99.

Dent, M. and Marinov, M. (2017) 'Introducing Automated Obstacle Detection to British Level Crossings', in *RailExchange Conference*. Available at: https://eprint.ncl.ac.uk/file_store/production/240699/216296A5-6F05-4371-B04D-BC4B9D42A16B.pdf.

Dumoulin, V. and Visin, F. (2016) ‘A guide to convolution arithmetic for deep learning’. Available at: http://arxiv.org/abs/1603.07285.

Ebeling, C. E. (2004) *An introduction to reliability and maintainability engineering*.

Edvard (2014) *Purposes and Examples of Safety Interlocking Devices*, *Electrical Engineering Portal*. Available at: https://electrical-engineering-portal.com/purposes-and-examples-of-safety-interlocking-devices#:~:text='Two-out-of-,the busbar section circuit breaker.&text=Auxiliary switches are fitted within,when all three are closed. (Accessed: 5 October 2020).

Elgammal, A., Harwood, D. and Davis, L. (2000) ‘Non-parametric Model for Background Substraction’, in *Computer Vision—ECCV 2000*. Berlin, Heidelberg: Springer, pp. 751–767. Available at: http://www.springerlink.com/index/3mcvhnwfa8bj4ln5.pdf%5Cnhttp://link.springer.com/chapter/10.1007/3-540-45053-X_48.

Endersby, T. (2016) ‘Viability of ETCS limited supervision for GB application : high-level study’, *RSSB*, (June 2015). doi: 10.13140/RG.2.1.2394.9208.

Erhan, D. *et al.* (2014) ‘Scalable Object detection using deep neural networks’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2142154. doi: 10.1109/ICCONS.2017.8250570.

European Union Agency for Railways (2017) *Railway Safety in the European Union - Safety overview 2017*. doi: 10.2821/474487.

Evans, A. W. and Hughes, P. (2019) ‘Traverses , delays and fatalities at railway level crossings in Great Britain’, *Accident Analysis and Prevention*, 129(May), pp. 66–75. doi: 10.1016/j.aap.2019.05.006.

Fakhfakh, N. *et al.* (2011) ‘A Video-Based Object Detection System for Improving Safety at Level Crossings’, *The Open Transportation Journal*, 5, pp. 45–59. doi: 10.2174/1874447801105010045.

Felzenszwalb, P. F. *et al.* (2009) ‘Object Detection with Discriminatively Trained Part Based Model’, *IEEE transactions on pattern analysis and machine intelligence*, 32(9), pp. 1627–1645.

Felzenszwalb, P. F. *et al.* (2010) ‘Object Detection with Discriminatively Trained Part Based Models’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9), p. 1627.

FLIR (2016) *Intelligent Transportation Systems, Detection and monitoring solutions for traffic and public transportation applications*, *FLIR.COM*. Available at: http://www.flirmedia.com/MMC/CVS/Traffic/IT_0004_EN.pdf.

FLIR (2017) *Thermal Imaging for Safety and Efficiency in Public Transportation*, *www.Flir.co.uk*. Available at: https://www.flir.co.uk/discover/traffic/public-transportation/thermal-imaging-for-safety-and-efficiency-in-public-transportation/ (Accessed: 26 February 2020).

Freund, Y. and Schapire, R. E. (1995) ‘A decision-theoretic generalization of on-line learning and an application to boosting’, in *In European conference on computational learning theory*. Berlin, Heidelberg: Springer, pp. 23–37. doi: 10.1007/3-540-59119-2_166.

Friedman, N. and Russell, S. (1996) ‘Image Segmentation in Video Sequences: A Probabilistic Approach’, pp. 175–181.

Fu, C.-Y. *et al.* (2017) ‘DSSD : Deconvolutional Single Shot Detector’. Available at: http://arxiv.org/abs/1701.06659.

Ganesan, V., Chitre, M. and Brekke, E. (2016) ‘Robust underwater obstacle detection and collision avoidance’, *Autonomous Robots*, 40(7), pp. 1165–1185. Available at: https://link.springer.com/article/10.1007/s10514-015-9532-2.

Garcia, C., Delakis, M. and Intelligence, M. (2004) ‘Convolutional Face Finder : A Neural Architecture for Fast and Robust Face Detection’, 26(11).

García, J. J. *et al.* (2010) ‘Sensory system for obstacle detection on high-speed lines’, *Transportation Research Part C: Emerging Technologies*, 18(4), pp. 536–553. Available at: http://dx.doi.org/10.1016/j.trc.2009.10.002.

Gaudet, Ch. (2016) *Using Binary Neural Networks for Hardware Branch Prediction*. doi: 10.13140/RG.2.1.1713.7521.

geograph (2014) *Level crossing cameras, Dunmurry*. Available at: https://www.geograph.ie/photo/4284271.

Géron, A. (2019) *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media.

Giannì, C. *et al.* (2017) ‘Obstacle detection system involving fusion of multiple sensor technologies’, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, p. 127. Available at: https://www.semanticscholar.org/paper/Obstacle-detection-system-involving-fusion-of-Gianni-Balsi/2c763a1bb73a6dab730f7163bda1a9c2deadd7a6.

Girshick, R. *et al.* (2014) ‘Rich feature hierarchies for accurate object detection and semantic segmentation’, *Proceedings of the CVPR*, pp. 2–9.

Girshick, R. (2015) ‘Fast R-CNN’, in *Proceedings of the IEEE international conference on computer vision (*, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

Glorot, X., Bordes, A. and Bengio, Y. (2011) ‘Deep Sparse Rectifier Neural Networks’, in *In Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323. Available at: http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. MIT Press.

Govoni, M. *et al.* (2015) ‘Ultra-Wide Bandwidth Systems for the Surveillance of Railway Crossing Areas’, *IEEE Communications Magazine*, pp. 117–123. Available at: https://ieeexplore.ieee.org/abstract/document/7295472.

Guo, Y. *et al.* (2016) ‘Deep Learning for Visual Understanding’, *Neurocomputing*, 187, pp. 27–48.

He, K. *et al.* (2014) ‘Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition’, *Proceedings of the ECCV*, pp. 1–14.

He, K. *et al.* (2015) ‘Delving deep into rectifiers: Surpassing human-level performance on imagenet classification’, in *In Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034. doi: 10.1109/ICCV.2015.123.

He, K. *et al.* (2016) ‘Deep residual learning for image recognition’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 770–778. doi: 10.1109/CVPR.2016.90.

Hilleary, T. N. and John, R. S. (2011) ‘Development and Testing of a Radar-Based Non-Embedded Vehicle Detection System for Four Quadrant Gate Warning Systems and Blocked Crossing Detection’, in *Arema 2011 Annual Conference*. Minneapolis.

Hinton, G. *et al.* (2012) ‘Improving neural networks by preventing co-adaptation of feature detectors’, pp. 1–18.

Hinton, G. E., Osindero, S. and Teh, Y.-W. (2006) ‘A Fast Learning Algorithm for Deep belief Nets’, *Neural Computation*, 18(7), pp. 1527–1554. doi: 10.1162/neco.2006.18.7.1527.

Hisamitsu, Y., Sekimoto, K., Nagata, K., Uehara, M., & Ota, E. (2008) ‘3-D laser radar level crossing obstacle detection system’, *IHI Engineering Review*, 41(2), pp. 51–57.

Hongchenzimo (2018) *Deeplearning - Overview of Convolution Neural Network*.

Horne, D. *et al.* (2016) ‘Evaluation of radar vehicle detection at four quadrant gate rail crossings’, *Journal of Rail Transport Planning & Management*, 6(2), pp. 149–162. Available at: https://pdf.sciencedirectassets.com/280481/1-s2.0-S2210970616X00042/1-s2.0-S2210970616300063/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEN3%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJGMEQCIF3bGKYpsAroj8%2B8wg7FK0OyRQNvG7ijPnyjGlCAHe4hAiBFItZ4RTqv5b.

How, F. (2020) *Back to basics: Interlocking Part 1*, *Institution of Railway Signal Engineers*.

Howard, A. G. *et al.* (2017) ‘MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications’. Available at: http://arxiv.org/abs/1704.04861.

Hsieh, H. H. *et al.* (2015) ‘Appling Lidar-based obstacle detection and wireless image transmission system for improving safety at level crossings’, *In 2015 International Carnahan Conference on Security Technology (ICCST)*, pp. 363–367. Available at: https://ieeexplore.ieee.org/abstract/document/7389711.

Hu, W. *et al.* (2013) ‘Incremental tensor subspace learning and Its applications to foreground segmentation and tracking’, *International Journal of Computer Vision*, 91(3), pp. 303–327. Available at: https://link.springer.com/article/10.1007/s11263-010-0399-6.

Huang, G. *et al.* (2017) ‘Densely connected convolutional networks’, *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.

ImageNet (2020) *ImageNet*, *Stanford Vision Lab*. Available at: http://www.image-net.org/ (Accessed: 20 May 2020).

Ioffe, S. and Szegedy, C. (2015) ‘Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift’.

Jia, Y. *et al.* (2014) ‘Caffe : Convolutional Architecture for Fast Feature Embedding’, *ACM MM*.

Jones, D., Vine, K. and McManus, M. (2018) *Product specification for an obstacle detection system at level crossings*.

JU, H. *et al.* (2011) ‘Integrating RAMS approach on the safety life cycle of rail transit’, in *In 2011 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*. IEEE, pp. 801–803.

Junghans, M. *et al.* (2016) ‘Wide-area based traffic situation detection at an ungated level crossing’, *International Journal of Safety and Security Engineering*, 6(2), pp. 383–393. doi: 10.2495/SAFE-V6-N2-383-393.

Karakose, M., Akın, E. and Tastimur, C. (2017) ‘Image Processing Based Level Crossing Detection and Foreign Objects Recognition Approach in Railways’, *International Journal of Applied Mathematics, Electronics and Computers*, 1(SpecialIssue), pp. 19–23. doi: 10.18100/ijamec.2017SpecialIssue30465.

Karpathy, A. (2018) *Convolutional Neural Networks for Visual Recognition*.

Kim, G. *et al.* (2012) ‘Design of safety equipment for railroad level crossings using laser range finder’, in *9th International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, pp. 2909–2913. doi: 10.1109/FSKD.2012.6234334.

Krizhevsky, A. and Hinton, G. E. (2012) ‘ImageNet Classification with Deep Convolutional Neural Networks’, *Advances in neural information processing systems*, pp. 1097–1105.

Kuo, J. C. C. (2016) ‘Understanding convolutional neural networks with a mathematical model’, *Journal of Visual Communication and Image Representation*, 41, pp. 406–413. doi: 10.1016/j.jvcir.2016.11.003.

Landsberg, P. (2014) *Thermodynamics and Statistical Mechanics*. NewYork: Dovers Publication. Available at: https://books.google.co.uk/books?hl=en&lr=&id=NaQ-AwAAQBAJ&oi=fnd&pg=PP1&dq=Thermodynamics+and+Statistical+Mechanics+by+Peter+Landsber&ots=JV5sAfF-py&sig=vTvsM6INII_iwdnGPmAQa0b0e6k&redir_esc=y#v=onepage&q=Thermodynamics and Statistical Mechanics by Peter.

Le, Q. V *et al.* (2010) ‘Tiled convolutional neural networks’, *Proceedings of the NIPS*, pp. 1–9.

Lecun, Y. *et al.* (1998) ‘Gradient-Based Learning Applied to Document Recognition’, (November), pp. 1–46.

LeCun, Y. *et al.* (1990) ‘Handwritten Digit Recognition with a Back-Propagation Network’, *In Advances in neural information processing systems*, pp. 396–404. doi: 10.4324/9781351195553-1.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) ‘Deep Learning’, *nature*, 521(7553), pp. 436–444. Available at: https://www.nature.com/articles/nature14539.

Leddar, T. (2018) *SOLID-STATE LiDARS : ENABLING THE AUTOMOTIVE INDUSTRY Towards Autonomous Driving*, *Leddar Tech*. Available at: https://www.tu-auto.com/intelligence/solid-state-lidars-enabling-the-automotive-industry-towards-autonomous-driving/.

Leonardis, A. (2002) *Subspace Methods for Visual Learning and Recognition*, *Faculty of Computer and Information Science*. Available at: papers3://publication/uuid/F61BA0CC-4EF2-4B55-AFC1-8FCB13C40679.

Li, Y. *et al.* (2003) ‘An integrated algorithm of incremental and robust PCA’, in *In Proceedings 2003 International Conference on Image Processing*. IEEE, pp. 1–245. doi: 10.1109/ICIP.2003.1246944.

Lienhart, R. and Maydt, J. (2002) ‘An Extended Set of Haar-like Features for Rapid Object Detection Rainer’, *In Proceedings. international conference on image processing*, 1. doi: 10.1016/0370-2693(92)90806-F.

Lin, T. *et al.* (2017) ‘Feature pyramid network for object detection’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 428–431. doi: 10.1109/ICVRIS.2019.00110.

Lipton, A. J., Fujiyoshi, H. and Patil, R. S. (1998) 'Moving Target Classification and Tracking from Real-time Video', in *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98*. IEEE, pp. 8–14.

Long, J., Shelhamer, E. and Darrell, T. (2015) 'Fully Convolutional Networks for Semantic Segmentation', *Proceedings of the CVPR*.

Low, D. G. (2004) 'Distinctive image features from scale-invariant keypoints', *International Journal of Computer Vision*, 60(2), pp. 91–110. Available at: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf.

Manduchi, R. *et al.* (2005) 'Obstacle detection and terrain classification for autonomous off-road navigation', *Autonomous Robots*, 18(1), pp. 81–102. Available at: https://link.springer.com/article/10.1023/B:AURO.0000047286.62481.1d.

Manikandan R, B. M. and S, P. (2017) 'Vision based obstacle detection on railway track', *International Journal of Pure and Applied Mathematics*, 116(24), pp. 567–576.

Masaki Yamazaki, Gang Xu, Y.-W. C. (2006) 'Detection of Moving Objects by Independent Component Analysis', in *Asian Conference on Computer Vision*. Berlin, Heidelberg: Springer. Available at: https://link.springer.com/chapter/10.1007/11612704_47.

Matlab (2020a) *Pedestrian and Bicyclist Classification Using Deep Learning*, *Matlab*. Available at: https://uk.mathworks.com/help/deeplearning/ug/pedestrian-and-bicyclist-classification-using-deep-learning.html (Accessed: 20 May 2020).

Matlab (2020b) *What Is Deep Learning? 3 things you need to know*, *Matlab*. Available at: https://uk.mathworks.com/discovery/deep-learning.html..html (Accessed: 26 February 2020).

Milutinović, D. and Lučanin, V. (2005) 'Relation between reliability and availability of railway vehicles', *FME Transactions,* 33, pp. 135–139.

Minsky, M. and Seymour, P. A. (2017) *Perceptrons: An introduction to computational geometry*. MIT Press.

Ml-cheatsheet.readthedocs.io (2017) *Loss functions*, *ml-cheatsheet*. Available at: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html (Accessed: 15 May 2020).

Moubray, J. (2001) *Reliability-centered maintenance*. Industrial Press Inc.

Mun, J., Kim, H. and Lee, J. (2018) 'A Deep Learning Approach for Automotive Radar Interference Mitigation', in *IEEE Vehicular Technology Conference*. IEEE, pp. 1–5. doi: 10.1109/VTCFall.2018.8690848.

Nair, V. and Hinton, G. E. (2010) 'Rectified Linear Units Improve Restricted Boltzmann Machines', in *Proceedings of the 27th international conference on machine learning*, pp. 807–814. doi: 10.1123/jab.2016-0355.

Network Rail (2012) *Signalling Design: Module X02 – Level Crossings: Common Design Requirements*.

Network Rail (2018a) *Catalogue of Network Rail Standards*. Available at: https://www.networkrail.co.uk/wp-content/uploads/2018/12/NR_CAT_STP_001-Issue-110.pdf.

Network Rail (2018b) 'Network Rail Telecom Strategic Plan', (January). Available at: https://www.networkrail.co.uk/wp-content/uploads/2018/02/Telecoms-Strategic-Plan.pdf.

Network Rail (2019a) *Level crossing events*. Available at: https://www.networkrail.co.uk/who-we-are/how-we-work/performance/safety-performance/level-crossing-events/ (Accessed: 4 December 2019).

Network Rail (2019b) *Our Approach to Managing Level Crossing Safety*. Available at: https://www.n-kesteven.gov.uk/_resources/assets/attachment/full/0/4300.pdf.

Network Rail (2020a) *Crossrail*, *Network Rail*. Available at: https://www.networkrail.co.uk/running-the-railway/railway-upgrade-plan/key-projects/crossrail (Accessed: 14 April 2020).

Network Rail (2020b) *Great North Rail Project*, *Network Rail*. Available at: https://www.networkrail.co.uk/running-the-railway/railway-upgrade-plan/key-projects/great-north-rail-project (Accessed: 14 April 2020).

Network Rail (2020c) *LX Sharing*, *Level Crossing knowledge Hub*. Available at: https://s6.newzapp.co.uk/t/gtp/OSwxNDIxODAwMzA2LDM=/ (Accessed: 21 April 2020).

Network Rail (2020d) *Our Delivery Plan for 2019-2024*. Available at: https://www.networkrail.co.uk/who-we-are/publications-and-resources/our-delivery-plan-for-2019-2024/#downloadall (Accessed: 14 April 2020).

Ng, A. (2016) *Machine learning Yearning*, *Studies in Systems, Decision and Control*. deeplearning.ai. doi: 10.1007/978-981-10-1509-0_9.

Ng, A. (2018) *Deep Leanring Yearning*. deeplearning.ai.

Nicholls, D. (2005) *System reliability toolkit*. Riac.

Nielsen, M. (2019a) ‘How the backpropagation algorithm works?’, in *Neural Networks and Deep Learning*. Available at: http://neuralnetworksanddeeplearning.com/chap2.html.

Nielsen, M. (2019b) ‘Using neural nets to recognize handwritten digits’, in *Neural Networks and Deep Learning*. Available at: http://neuralnetworksanddeeplearning.com/chap1.html.

Noh, H., Hong, S. and Han, B. (2015) ‘Learning deconvolution network for semantic segmentation’, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1520–1528. doi: 10.1109/ICCV.2015.178.

O’Donnell, R. M. (2007) *Introduction to RADAR Systems*, *MIT Open Courseware*.

Office of Rail and Road (2011) *Level crossings : A guide for managers, designers and operators Railway Safety Publication*, *Regulation*. Available at: https://orr.gov.uk/__data/assets/pdf_file/0016/2158/level_crossings_guidance.pdf.

Office of Rail and Road (2016) *ORR’s Annual Health and Safety Report of Performance on Britain’s Railways: 2015-16*. Available at: https://orr.gov.uk/__data/assets/pdf_file/0020/22457/annual-health-and-safety-report-july-2016.pdf.

Office of Rail and Road (2017) *Annual Safety Performance Report. A reference guide to safety trends on GB railways 2016/17*. doi: 10.1210/jc.2013-2845.

Ohta, M. (2005) *Level Crossings Obstacle Detection System Using Stereo Cameras*, *Quarterly Report of RTRI*. doi: 10.2219/rtriqr.46.110.

Oquab, M. (2012) ‘Is object localization for free ? – Weakly-supervised learning with convolutional neural networks’, (iii).

Osadchy, M. (2007) ‘Synergistic Face Detection and Pose Estimation with Energy-Based Models’, 8, pp. 1197–1215.

Ouyang, W. *et al.* (2015) ‘DeepID-Net : Deformable Deep Convolutional Neural Networks for Object Detection’, *Proceedings of the CVPR*.

Park, M. G. (2014) *RAMS management of railway systems*. Birmingham. Available at: http://etheses.bham.ac.uk/4750/.

Patel, H. A. and Tank, P. M. (2015) ‘Survey on Moving Object Detection Techniques’, *International Journal of Computer Science and Mobile Computing*, 4(6), pp. 858–861.

Pavlović, M. G. *et al.* (2018) ‘Advanced thermal camera based system for object detection on rail tracks’, *Thermal Science*, 22, pp. S1551–S1561. doi: 10.2298/TSCI18S5551P.

PAVLOVIĆ, M., PAVLOVIĆ, N. T. and PAVLOVIĆ, V. (2016) ‘Methods for Detection of Obstacles on the Railway Level Crossing’, in *Scientific-Expert Conference on Railways RAILCON ‘16*, pp. 121–124.

Petrov, S. (2011) ‘Loop detectors in active Level Crossing applications’, *AusRAIL PLUS*, pp. 22–24. Available at: http://railknowledgebank.com/Presto/content/GetDoc.axd?ctID=MTk4MTRjNDUtNWQ0My00OTBmLTllYWUtZWFjM2U2OTE0ZDY3&rID=MjE3OA==&pID=Nzkx&attchmnt=True&uSesDM=False&rIdx=MTcwOA==&rCFU=.

Piccardi, M. (2004) ‘Background subtraction techniques: A review’, in *IEEE International Conference on Systems, Man and Cybernetics Background*. IEEE, pp. 3099–3104. doi: 10.1109/ICSMC.2004.1400815.

Pu, Y.-R., Chen, L.-W. and Lee, S.-H. (2014) ‘Study of Moving Obstacle Detection at Railway Crossing by Machine Vision’, *Information Technology Journal*, 13(16), pp. 2611–2618.

Railsigns (2020) *Automatic Warning System (AWS)*, *www.railsigns.uk*. Available at: http://www.railsigns.uk/info/aws1/aws1.html (Accessed: 2 October 2020).

Rajan, V. (2017) ‘IJARCCE Towards Efficient Intrusion Detection using Deep Learning Techniques: A Review’, *International Journal of Advanced Research in Computer and Communication Engineering ISO*, 6(10), pp. 375–384. doi: 10.17148/IJARCCE.2017.61066.

Redmon, J. *et al.* (2016) ‘You Only Look Once: Unified, Real-Time Object Detection’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Redmon, J. and Farhadi, A. (2017) ‘YOLO9000: Better, faster, stronger’, *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 6517–6525. doi: 10.1109/CVPR.2017.690.

Redmon, J. and Farhadi, A. (2018) ‘YOLOv3: An Incremental Improvement’. Available at: http://arxiv.org/abs/1804.02767.

Ren, S., He, K. and Girshick, R. (2015) ‘Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks’, *Advances in neural information processing systems*, pp. 91–99.

Ricco, J. (2017) *What is max pooling in convolutional neural networks?*, *Quora*. Available at: https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks (Accessed: 29 April 2020).

Roberts, N. (2020) *LIDAR Obstacle Detection*, *LBFoster*. Available at: https://ieeexplore-ieee-org.mmu.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=683593.

Rosenblatt, F. (1957) *The perceptron, a perceiving and recognizing automaton Project Para*.

Rosenblatt, F. (1958) 'The perceptron: a probabilistic model for information storage and organization in the brain', *Psychological review*, 65(6), p. 386. Available at: http://www2.fiit.stuba.sk/~cernans/nn/nn_texts/neuronove_siete_priesvitky_02_Q.pdf.

Ruck, D. W. *et al.* (1990) 'The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function', *IEEE Transactions on Neural Networks*, 1(4), p. 291. doi: 10.1109/72.80266.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) 'Learning representations by back-propagating errors', *Nature*, 323(6088), pp. 533–536. doi: 10.1038/323533a0.

Russakovsky, O. *et al.* (2015) 'ImageNet Large Scale Visual Recognition Challenge', *Int Journal Computer Vision*, 115, pp. 211–252.

Salmane, H., Khoudour, L. and Ruichek, Y. (2013) 'Improving safety of level crossings by detecting hazard situations using video based processing', in *IEEE International Conference on Intelligent Rail Transportation Proceedings*. Beijing, pp. 179–184. doi: 10.1109/ICIRT.2013.6696290.

Salmane, H., Khoudour, L. and Ruichek, Y. (2016) 'Motion-based object tracking method for safety at level crossing', *Journal of Electronic Imaging*, 25(5).

Sato, K. *et al.* (1998) 'Obstruction detector using ultrasonic sensors for upgrading the safety of a level crossing', in *International Conference on Developments in Mass Transit Systems*. doi: 10.1049/cp:19980140.

Scherer, D., Andreas, M. and Behnke, S. (2010) 'Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition', *20th International Conference on Artificial Neural Networks (ICANN)*, (September).

Schiopu, D. (2009) 'Using Artificial Neural Networks in a Pattern Recognition Control System', *Petroleum-Gas University of Ploiesti Bulletin, Technical Series*, 6(13).

Schmidhuber, J. (2014) *Deep Learning in Neural Networks : An Overview*.

Seki, M. *et al.* (2003) 'Background subtraction based on cooccurrence of image variations', in *In 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. II–II. Available at: http://ieeexplore.ieee.org/document/1211453/.

Sharma, A. (2018) *What is the difference between CNN and a support vector machine?*, *Quora*. Available at: https://www.quora.com/What-is-the-difference-between-CNN-and-a-support-vector-machine (Accessed: 26 April 2020).

Shetty, R. *et al.* (2019) 'Automated Railway Crossing and Obstacle Detection', in *International Conference on Advances in Science & Technology*. Mumbai. doi: 10.2139/ssrn.3367991.

Šilar, Z. and Dobrovolný, M. (2013) 'The obstacle detection on the railway crossing based on optical flow and clustering', in *36th International Conference on Telecommunications and Signal Processing (TSP)*. Rome, pp. 755–759. doi: 10.1109/TSP.2013.6614039.

Simonyan, K. and Zisserman, A. (2015) 'VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE

IMAGE RECOGNITION', in *Proceedings of the ICLR*, pp. 1–14.

Skalski, P. (2019) *Gentle Dive into Math Behind Convolutional Neural Networks*, *Towards Data Science*. Available at: https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9.

Skolnik, M. I. (2020) 'RADAR', *Encyclopædia Britannica*. Encyclopædia Britannica, inc. Available at: https://www.britannica.com/technology/radar.

Spowart, F. M. (2014) *Level Crossing CCTV notes*.

Srivastava, N. *et al.* (2014) 'Dropout : A Simple Way to Prevent Neural Networks from Overfitting', *Journal of Machine Learning Research*, 15, pp. 1929–1958.

Stauffer, C. and Grimson, W. E. L. (1999) 'Adaptive background mixture models for real-time tracking', *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 2, pp. 246–252. doi: 10.1109/CVPR.1999.784637.

Steven, A. (2014) *Level Crossings The Law Commission and The Scottish Law Commission*. Available at: https://www.scotlawcom.gov.uk/law-reform/law-reform-projects/completed-projects/level-crossings/.

Sun, Y., Wang, X. and Tang, X. (2013) 'Deep Convolutional Network Cascade for Facial Point Detection', *Proceedings of the CVPR*, pp. 0–7. doi: 10.1109/CVPR.2013.446.

Szegedy, C. *et al.* (2015) 'Going Deeper with Convolutions', in *In Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tan, M. and Le, Q. V. (2019) 'EfficientNet: Rethinking model scaling for convolutional neural networks', in *36th International Conference on Machine Learning, ICML 2019*, pp. 10691–10700.

Tan, M., Pang, R. and Le, Q. V. (2020) 'EfficientDet: Scalable and Efficient Object Detection', in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790. doi: 10.1109/cvpr42600.2020.01079.

Tax, D. M. J. and Duin, R. P. W. (2004) 'Support Vector Data Description', *Machine Learning*, 54(1), pp. 45–66. Available at: https://link.springer.com/article/10.1023/B:MACH.0000008084.60811.49.

Tompson, J. *et al.* (2015) 'Efficient Object Localization Using Convolutional Networks', *In Proceedings Conference on Computer Vision and Pattern Recognition*.

Truong, P. (2019) *Loss functions: Why, what, where or when?*, *Medium*. Available at: https://medium.com/@phuctrt/loss-functions-why-what-where-or-when-189815343d3f (Accessed: 15 May 2020).

Tsang, S.-H. (2018) *Review: Faster RCNN (Object Detection)*, *Medium-Towards Data Science*. Available at: https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202 (Accessed: 5 May 2020).

Turaga, S. C., Murray, J. F. and Seung, H. S. (2010) 'Convolutional Networks Can Learn to Generate Affinity', 538, pp. 511–538.

Uijlings, J. R. R. *et al.* (2013) 'Selective search for object recognition', *International Journal of Computer Vision*, 104(2), pp. 154–171. doi: 10.1007/s11263-013-0620-5.

Vaillant, R., Monrocq, C. and Cun, Y. Le (1994) 'Original approach for the localisation of objects in images', *IEE Proceedings - Vision, Image and Signal Processing*, 141(4), pp. 245–250. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=318027&isnumber=7665.

Valera, M. and A. Velastin, S. (2005) 'Intelligent distributed surveillance systems: a review', in *IEEE Proceedings-Vision, Image and Signal Processing*, pp. 192–204.

Vintr, Z. and Vintr, M. (2007) *Reliability and Safety of Rail Vehicle Electromechanical Systems*.

Wager, S., Wang, S. and Liang, P. (2013) 'Dropout Training as Adaptive Regularization', *Proceedings of the NIPS*, pp. 1–9.

Wan, L. *et al.* (2012) 'Regularization of Neural Networks using DropConnect', *Proceedings of the ICMl*, (1).

Wang, J., Bebis, G. and Miller, R. (2006) 'Robust video-based surveillance by integrating target detection with tracking', in *n 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*. IEEE, p. 137. Available at: https://ieeexplore.ieee.org/abstract/document/1640582.

Wang, N., Choi, J. and Gopalakrishnan, K. (2018) *8-Bit Precision for Training Deep Learning Systems*, *IBM*. Available at: https://www.ibm.com/blogs/research/2018/12/8-bit-precision-training/ (Accessed: 26 February 2020).

Wang, S. I. and Manning, C. D. (2013) 'Fast dropout training', *Proceedings of the ICML*, 28.

Wang, X. *et al.* (2014) 'Deep Joint Task Learning for Generic Object Extraction', *Proceedings of the NIPS*, pp. 1–9.

Warde-farley, D. *et al.* (2014) 'An empirical analysis of dropout in piecewise linear networks', Prceedings, pp. 1–10.

Wolff, C. (2020) *Range or distance measurement*, *radartutotial.eu*.

Woolford, P. (2002) *Guidance on Provision , Risk Assessment and Review of Level Crossings*. London.

Wren, C. R. *et al.* (1997) 'Pfinder: Real-Time Tracking of the Human Body', *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), pp. 780–785. Available at: https://ieeexplore.ieee.org/abstract/document/598236.

Wu, J. (2017) 'Introduction to Convolutional Neural Networks', *National Key Lab for Novel Software Technology*, 5, p. 23. doi: 10.1007/978-1-4842-5648-0.

Xu, B. *et al.* (2015) 'Empirical Evaluation of Rectified Activations in Convolutional Network'. Available at: http://arxiv.org/abs/1505.00853.

Yang, Z. and Nevatia, R. (2016) 'A Multi-Scale Cascade Fully Convolutional', *International Conference on Pattern Recognition (ICPR)*, 23, pp. 622–627.

Zeiler, M. D. and Fergus, R. (2014) 'Visualizing and Understanding Convolutional Networks', *Proceedings of ECCV*, pp. 818–833.

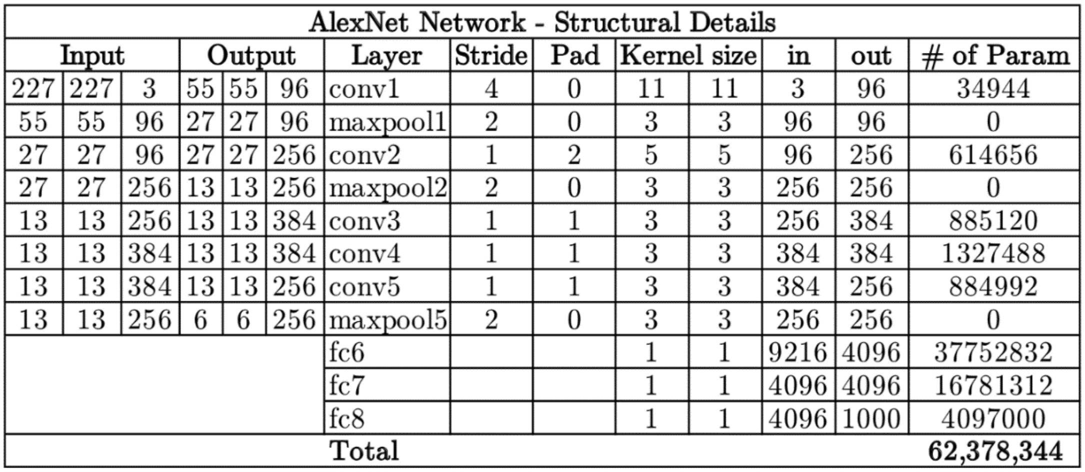
Zhao, Z. and Zheng, P. (2012) 'Object Detection with Deep Learning : A Review', *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS FOR PUBLICATION*, pp. 1–21.

Zhu, C. (2019) *The "Less is More" of Machine Learning*, *Towards Data Science*. Available at: https://towardsdatascience.com/the-less-is-more-of-machine-learning-1f571c0d4481 (Accessed: 15

May 2020).

8 APPENDIX

8.1 ALEXNET ARCHITECTURE

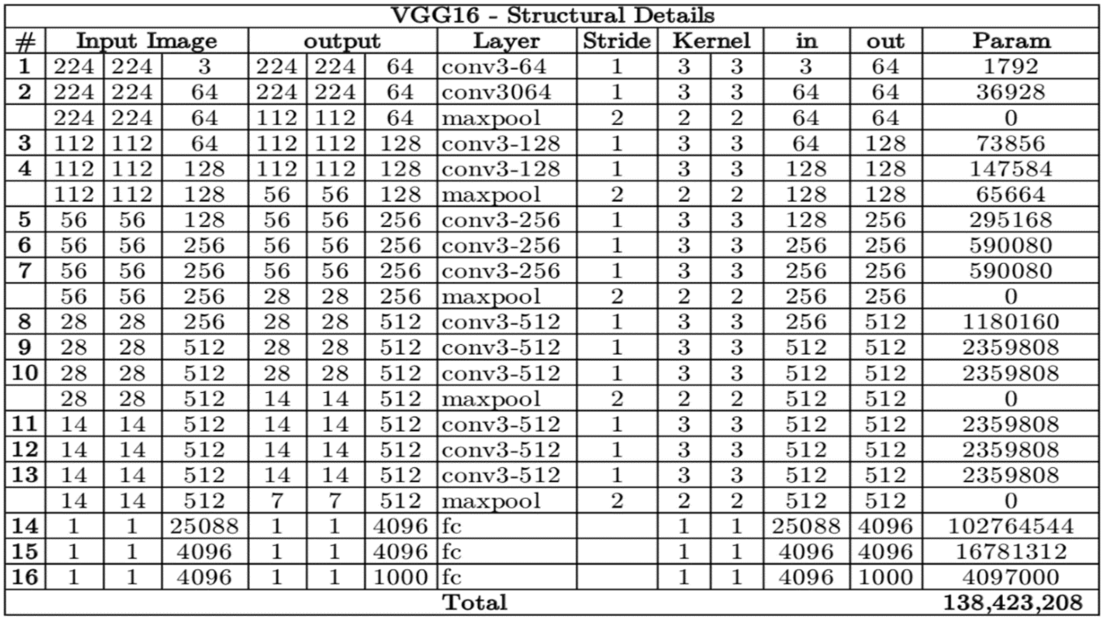


AlexNet Network - Structural Details

Input			Output			Layer	Stride	Pad	Kernel size		in	out	# of Param
227	227	3	55	55	96	conv1	4	0	11	11	3	96	34944
55	55	96	27	27	96	maxpool1	2	0	3	3	96	96	0
27	27	96	27	27	256	conv2	1	2	5	5	96	256	614656
27	27	256	13	13	256	maxpool2	2	0	3	3	256	256	0
13	13	256	13	13	384	conv3	1	1	3	3	256	384	885120
13	13	384	13	13	384	conv4	1	1	3	3	384	384	1327488
13	13	384	13	13	256	conv5	1	1	3	3	384	256	884992
13	13	256	6	6	256	maxpool5	2	0	3	3	256	256	0
						fc6			1	1	9216	4096	37752832
						fc7			1	1	4096	4096	16781312
						fc8			1	1	4096	1000	4097000
Total													**62,378,344**

Figure 43A detailed information about the AlexNet. The "conv" represents the Convolutional Layer, "fc" represents the Fully Connected Layer.

8.2 VGG-NET ARCHITECTURE

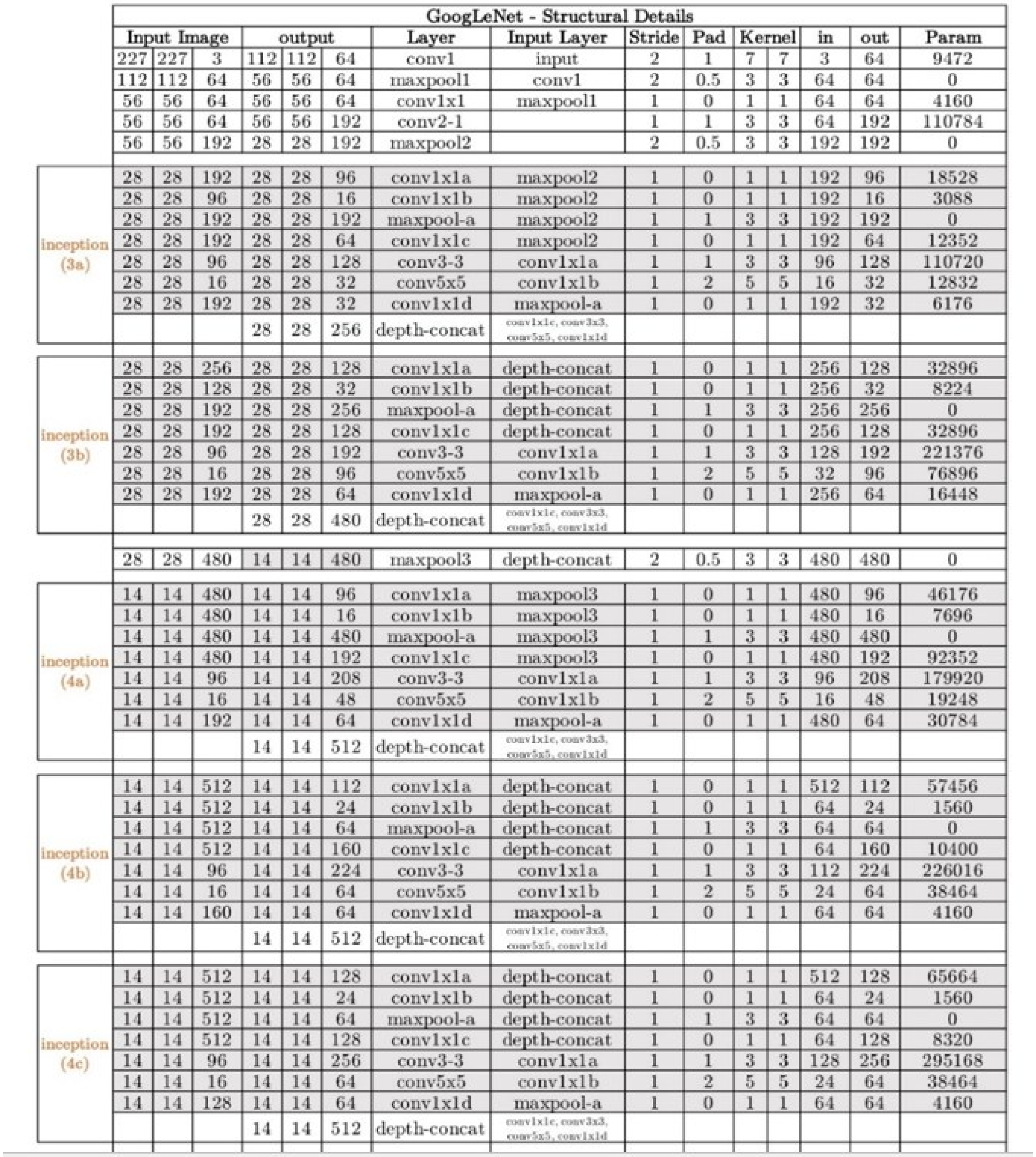


VGG16 - Structural Details

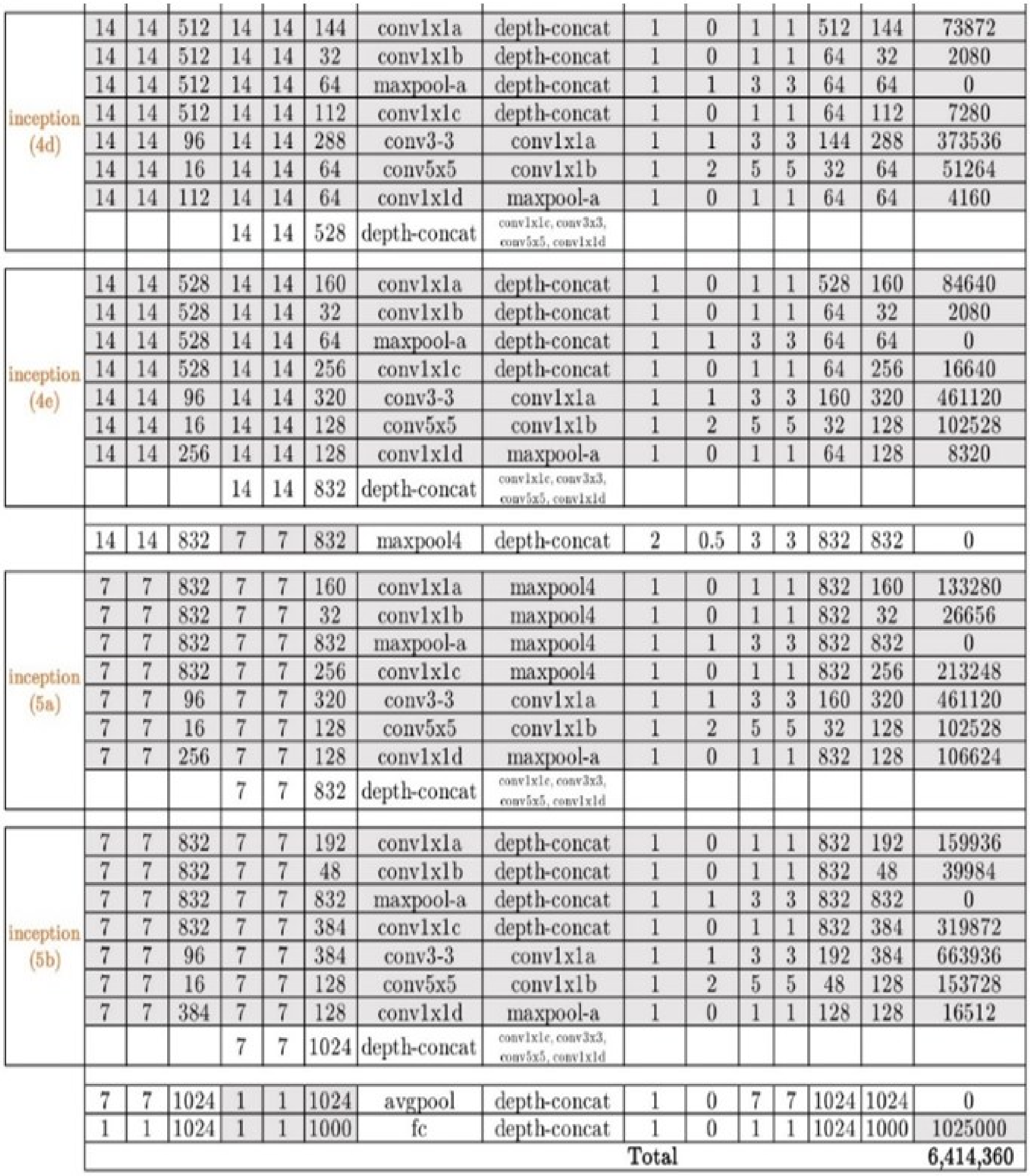
#	Input Image			output			Layer	Stride	Kernel		in	out	Param
1	224	224	3	224	224	64	conv3-64	1	3	3	3	64	1792
2	224	224	64	224	224	64	conv3064	1	3	3	64	64	36928
	224	224	64	112	112	64	maxpool	2	2	2	64	64	0
3	112	112	64	112	112	128	conv3-128	1	3	3	64	128	73856
4	112	112	128	112	112	128	conv3-128	1	3	3	128	128	147584
	112	112	128	56	56	128	maxpool	2	2	2	128	128	65664
5	56	56	128	56	56	256	conv3-256	1	3	3	128	256	295168
6	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
7	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
	56	56	256	28	28	256	maxpool	2	2	2	256	256	0
8	28	28	256	28	28	512	conv3-512	1	3	3	256	512	1180160
9	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
10	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
	28	28	512	14	14	512	maxpool	2	2	2	512	512	0
11	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
12	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
13	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
	14	14	512	7	7	512	maxpool	2	2	2	512	512	0
14	1	1	25088	1	1	4096	fc		1	1	25088	4096	102764544
15	1	1	4096	1	1	4096	fc		1	1	4096	4096	16781312
16	1	1	4096	1	1	1000	fc		1	1	4096	1000	4097000
Total													**138,423,208**

Figure 44A representation of VGG-Net architecture.

8.3 INCEPTION



	GoogLeNet - Structural Details														
	Input Image			output			Layer	Input Layer	Stride	Pad	Kernel		in	out	Param
	227	227	3	112	112	64	conv1	input	2	1	7	7	3	64	9472
	112	112	64	56	56	64	maxpool1	conv1	2	0.5	3	3	64	64	0
	56	56	64	56	56	64	conv1x1	maxpool1	1	0	1	1	64	64	4160
	56	56	64	56	56	192	conv2-1		1	1	3	3	64	192	110784
	56	56	192	28	28	192	maxpool2		2	0.5	3	3	192	192	0
inception (3a)	28	28	192	28	28	96	conv1x1a	maxpool2	1	0	1	1	192	96	18528
	28	28	96	28	28	16	conv1x1b	maxpool2	1	0	1	1	192	16	3088
	28	28	192	28	28	192	maxpool-a	maxpool2	1	1	3	3	192	192	0
	28	28	192	28	28	64	conv1x1c	maxpool2	1	0	1	1	192	64	12352
	28	28	96	28	28	128	conv3-3	conv1x1a	1	1	3	3	96	128	110720
	28	28	16	28	28	32	conv5x5	conv1x1b	1	2	5	5	16	32	12832
	28	28	192	28	28	32	conv1x1d	maxpool-a	1	0	1	1	192	32	6176
				28	28	256	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
inception (3b)	28	28	256	28	28	128	conv1x1a	depth-concat	1	0	1	1	256	128	32896
	28	28	128	28	28	32	conv1x1b	depth-concat	1	0	1	1	256	32	8224
	28	28	192	28	28	256	maxpool-a	depth-concat	1	1	3	3	256	256	0
	28	28	192	28	28	128	conv1x1c	depth-concat	1	0	1	1	256	128	32896
	28	28	96	28	28	192	conv3-3	conv1x1a	1	1	3	3	128	192	221376
	28	28	16	28	28	96	conv5x5	conv1x1b	1	2	5	5	32	96	76896
	28	28	192	28	28	64	conv1x1d	maxpool-a	1	0	1	1	256	64	16448
				28	28	480	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
	28	28	480	14	14	480	maxpool3	depth-concat	2	0.5	3	3	480	480	0
inception (4a)	14	14	480	14	14	96	conv1x1a	maxpool3	1	0	1	1	480	96	46176
	14	14	480	14	14	16	conv1x1b	maxpool3	1	0	1	1	480	16	7696
	14	14	480	14	14	480	maxpool-a	maxpool3	1	1	3	3	480	480	0
	14	14	480	14	14	192	conv1x1c	maxpool3	1	0	1	1	480	192	92352
	14	14	96	14	14	208	conv3-3	conv1x1a	1	1	3	3	96	208	179920
	14	14	16	14	14	48	conv5x5	conv1x1b	1	2	5	5	16	48	19248
	14	14	192	14	14	64	conv1x1d	maxpool-a	1	0	1	1	480	64	30784
				14	14	512	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
inception (4b)	14	14	512	14	14	112	conv1x1a	depth-concat	1	0	1	1	512	112	57456
	14	14	512	14	14	24	conv1x1b	depth-concat	1	0	1	1	64	24	1560
	14	14	512	14	14	64	maxpool-a	depth-concat	1	1	3	3	64	64	0
	14	14	512	14	14	160	conv1x1c	depth-concat	1	0	1	1	64	160	10400
	14	14	96	14	14	224	conv3-3	conv1x1a	1	1	3	3	112	224	226016
	14	14	16	14	14	64	conv5x5	conv1x1b	1	2	5	5	24	64	38464
	14	14	160	14	14	64	conv1x1d	maxpool-a	1	0	1	1	64	64	4160
				14	14	512	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
inception (4c)	14	14	512	14	14	128	conv1x1a	depth-concat	1	0	1	1	512	128	65664
	14	14	512	14	14	24	conv1x1b	depth-concat	1	0	1	1	64	24	1560
	14	14	512	14	14	64	maxpool-a	depth-concat	1	1	3	3	64	64	0
	14	14	512	14	14	128	conv1x1c	depth-concat	1	0	1	1	64	128	8320
	14	14	96	14	14	256	conv3-3	conv1x1a	1	1	3	3	128	256	295168
	14	14	16	14	14	64	conv5x5	conv1x1b	1	2	5	5	24	64	38464
	14	14	128	14	14	64	conv1x1d	maxpool-a	1	0	1	1	64	64	4160
				14	14	512	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							



inception (4d)	14	14	512	14	14	144	conv1x1a	depth-concat	1	0	1	1	512	144	73872
	14	14	512	14	14	32	conv1x1b	depth-concat	1	0	1	1	64	32	2080
	14	14	512	14	14	64	maxpool-a	depth-concat	1	1	3	3	64	64	0
	14	14	512	14	14	112	conv1x1c	depth-concat	1	0	1	1	64	112	7280
	14	14	96	14	14	288	conv3-3	conv1x1a	1	1	3	3	144	288	373536
	14	14	16	14	14	64	conv5x5	conv1x1b	1	2	5	5	32	64	51264
	14	14	112	14	14	64	conv1x1d	maxpool-a	1	0	1	1	64	64	4160
				14	14	528	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
inception (4e)	14	14	528	14	14	160	conv1x1a	depth-concat	1	0	1	1	528	160	84640
	14	14	528	14	14	32	conv1x1b	depth-concat	1	0	1	1	64	32	2080
	14	14	528	14	14	64	maxpool-a	depth-concat	1	1	3	3	64	64	0
	14	14	528	14	14	256	conv1x1c	depth-concat	1	0	1	1	64	256	16640
	14	14	96	14	14	320	conv3-3	conv1x1a	1	1	3	3	160	320	461120
	14	14	16	14	14	128	conv5x5	conv1x1b	1	2	5	5	32	128	102528
	14	14	256	14	14	128	conv1x1d	maxpool-a	1	0	1	1	64	128	8320
				14	14	832	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
	14	14	832	7	7	832	maxpool4	depth-concat	2	0.5	3	3	832	832	0
inception (5a)	7	7	832	7	7	160	conv1x1a	maxpool4	1	0	1	1	832	160	133280
	7	7	832	7	7	32	conv1x1b	maxpool4	1	0	1	1	832	32	26656
	7	7	832	7	7	832	maxpool-a	maxpool4	1	1	3	3	832	832	0
	7	7	832	7	7	256	conv1x1c	maxpool4	1	0	1	1	832	256	213248
	7	7	96	7	7	320	conv3-3	conv1x1a	1	1	3	3	160	320	461120
	7	7	16	7	7	128	conv5x5	conv1x1b	1	2	5	5	32	128	102528
	7	7	256	7	7	128	conv1x1d	maxpool-a	1	0	1	1	832	128	106624
				7	7	832	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
inception (5b)	7	7	832	7	7	192	conv1x1a	depth-concat	1	0	1	1	832	192	159936
	7	7	832	7	7	48	conv1x1b	depth-concat	1	0	1	1	832	48	39984
	7	7	832	7	7	832	maxpool-a	depth-concat	1	1	3	3	832	832	0
	7	7	832	7	7	384	conv1x1c	depth-concat	1	0	1	1	832	384	319872
	7	7	96	7	7	384	conv3-3	conv1x1a	1	1	3	3	192	384	663936
	7	7	16	7	7	128	conv5x5	conv1x1b	1	2	5	5	48	128	153728
	7	7	384	7	7	128	conv1x1d	maxpool-a	1	0	1	1	128	128	16512
				7	7	1024	depth-concat	conv1x1c, conv3x3, conv5x5, conv1x1d							
	7	7	1024	1	1	1024	avgpool	depth-concat	1	0	7	7	1024	1024	0
	1	1	1024	1	1	1000	fc	depth-concat	1	0	1	1	1024	1000	1025000
									Total						6,414,360

Figure 45Representation of Inception architecture.

8.4 RESULTS FROM IMAGE CLASSIFICATION USING TRANSFER LEARNING

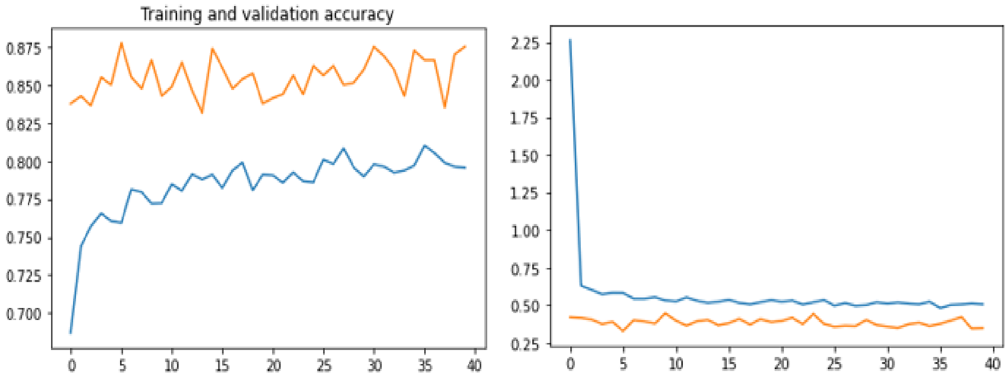


Figure 46Transfer Learning using DenseNet, where the training accuracy is about 79.55% (blue line) and Validation accuracy is 87.50% (orange line).

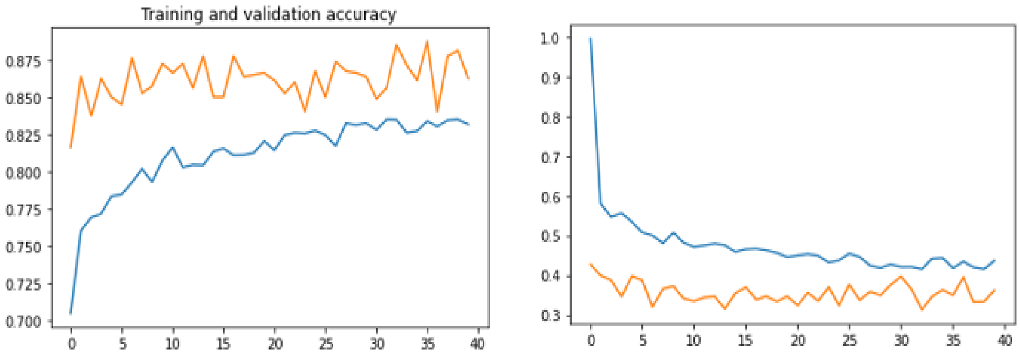


Figure 47Transfer Learning using Inception Network, where training accuracy is about 83.17% (blue line) and validation accuracy is about 86% (orange line).

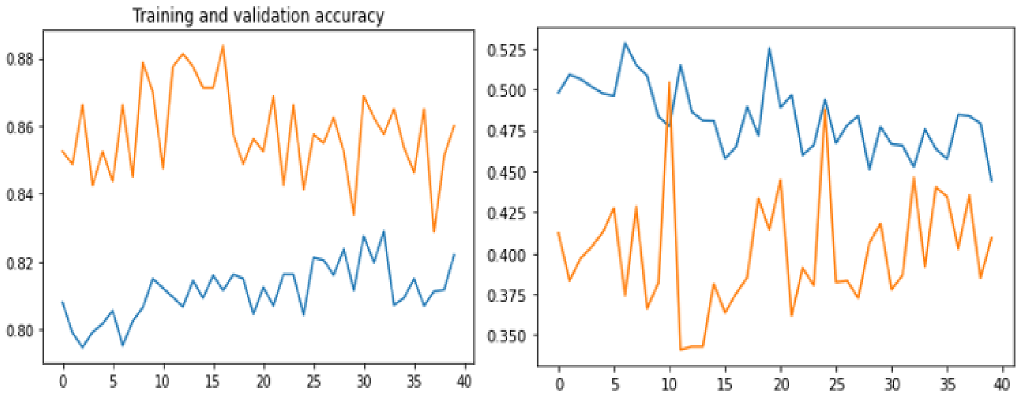


Figure 48Transfer Learning from Xception Network. The training accuracy is about 80% and validation accuracy is about 82%.

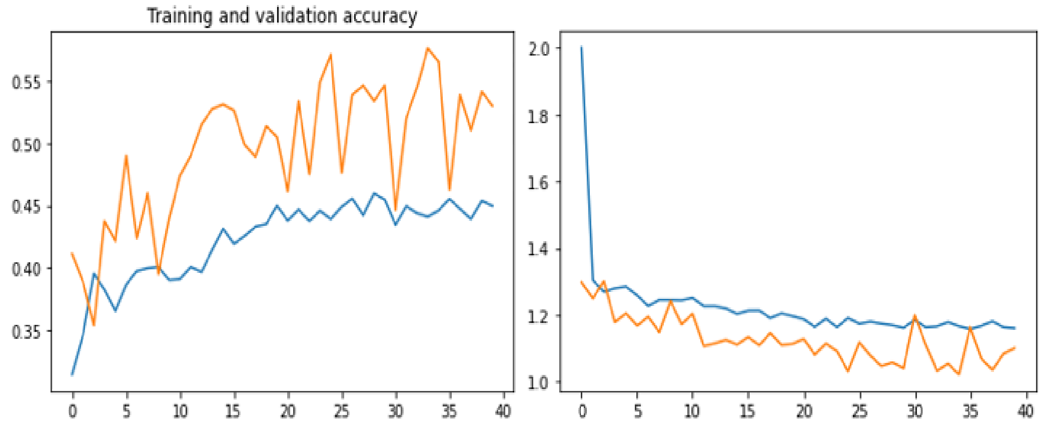


Figure 49Transfer Learning from ResNet, where training accuracy is about 44% (blue line) and validation accuracy is about 53% (orange line).

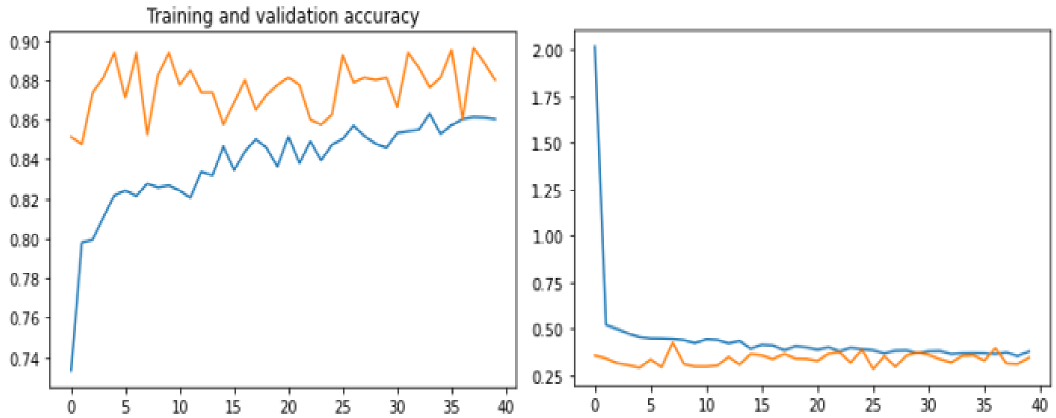


Figure 50Transfer Learning using MobileNet Network, where training accuracy is about 86% (blue line) and validation accuracy is about 88% (orange line).

8.5 Results from Object Detection Using Transfer Learning

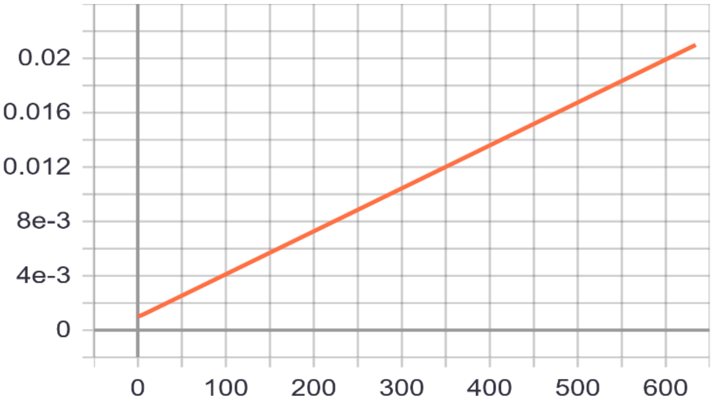


Figure 51Learning Rate for Efficient model trained using 25000 steps in Epochs. Visualization from tensorboard.

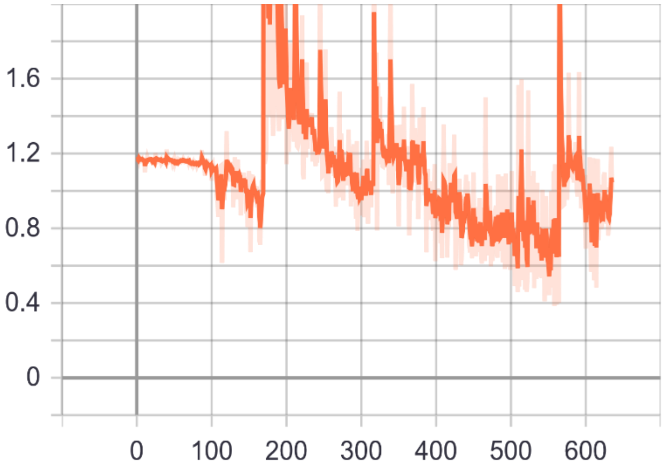


Figure 52Classification Loss for Efficient Model trained via Transfer Learning. Visualization from tensorboard.

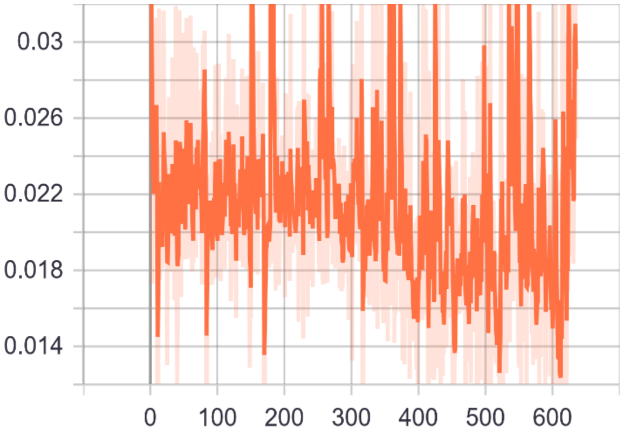


Figure 53Localization Loss for Efficient model during training. Visualization from tensorboard.

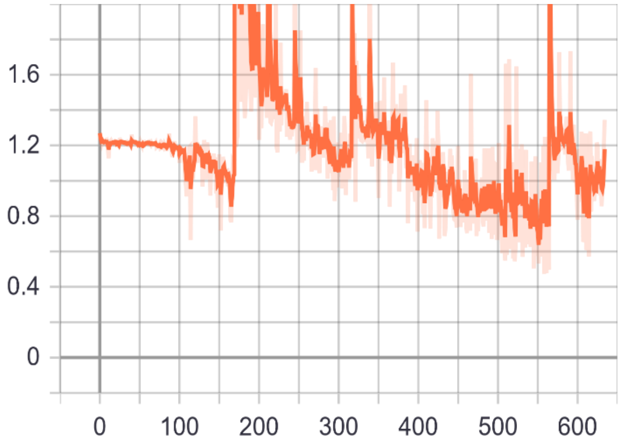


Figure 54Normalized total Loss from Efficient Model during its training. Visualization from tensorboard.

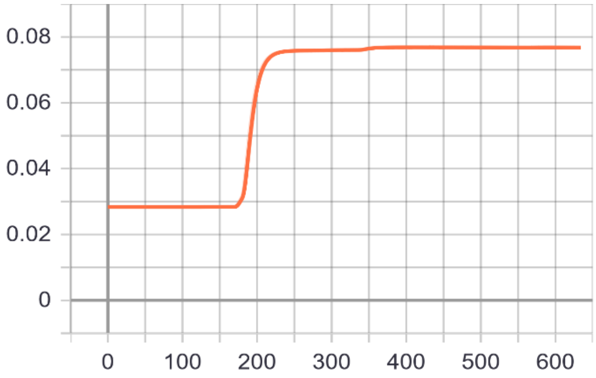


Figure 55Regularization Loss from Efficient model. Visualization from tensorboard.

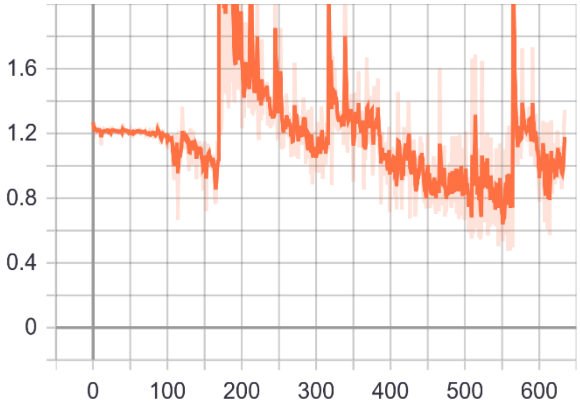


Figure 56Total Loss from Efficient model. Visualization from tensorboard.

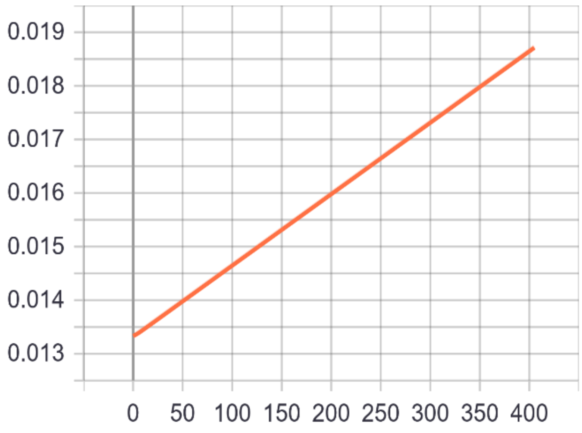


Figure 57Learning Rate for RCNN model trained using Transfer Learning. Visualization from tensorboard.

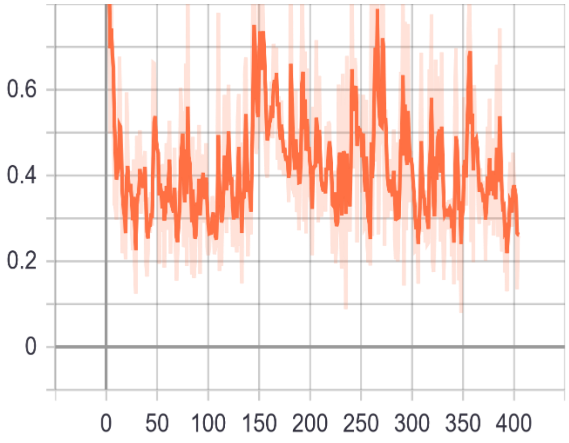


Figure 58Classification Loss from RCNN model. Visualization from tensorboard.

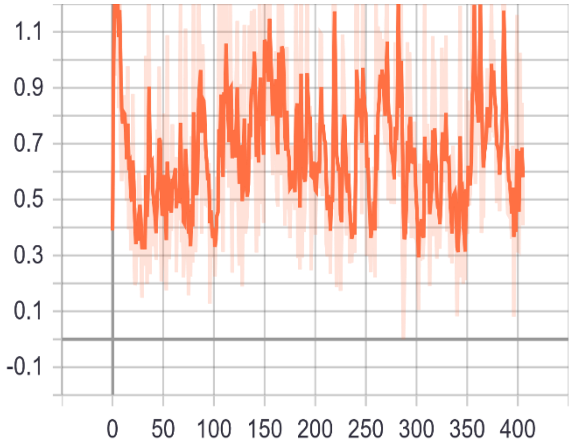


Figure 59Localization Loss from model RCNN. Visualization from tensorboard.

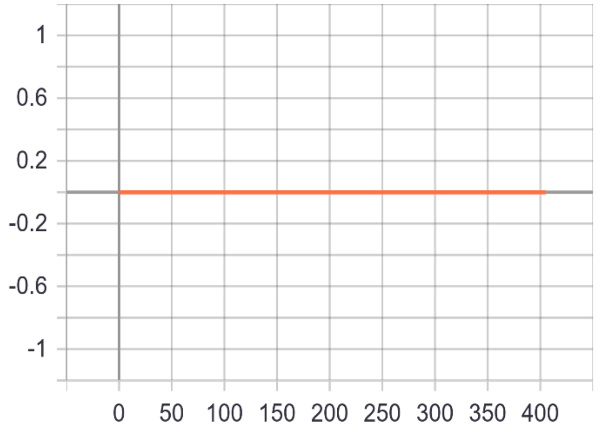


Figure 60Regularization Loss from model RCNN. Visualization from tensorboard.

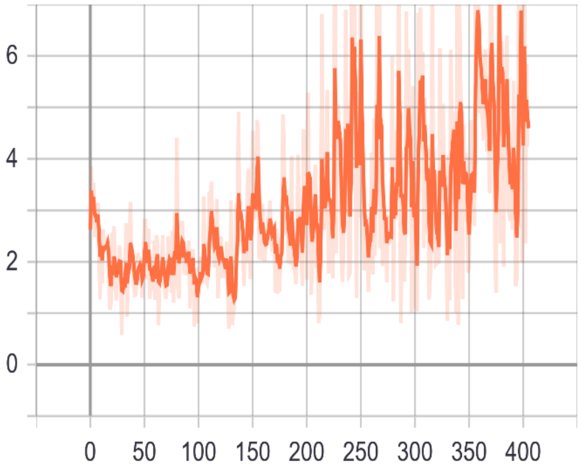


Figure 61Total Loss for model RCNN using transfer Learning techniques. Visualization from tensorboard.

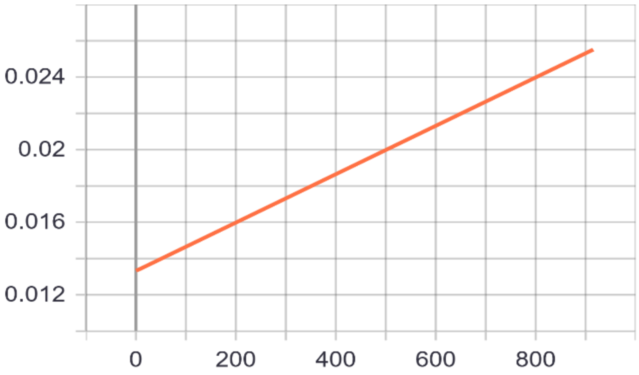


Figure 62Learning Rate for model ResNet trained using Transfer Learning techniques. Visualization from tensorboard.

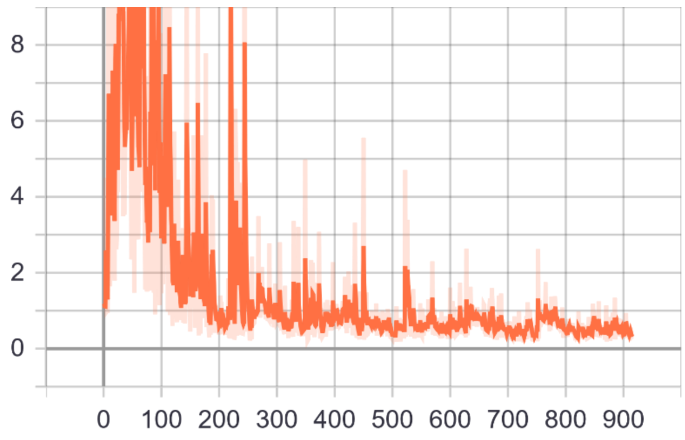


Figure 63Classification Loss from model ResNet. Visualization from tensorboard.

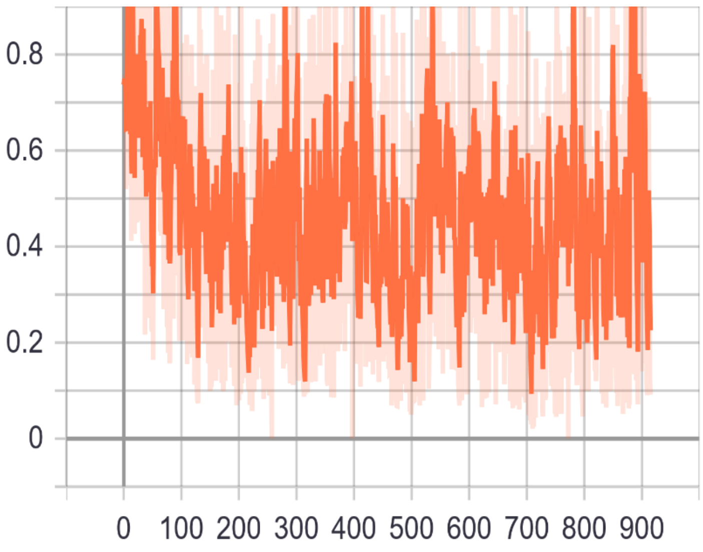


Figure 64Localisation Loss from model ResNet. Visualization from tensorboard.

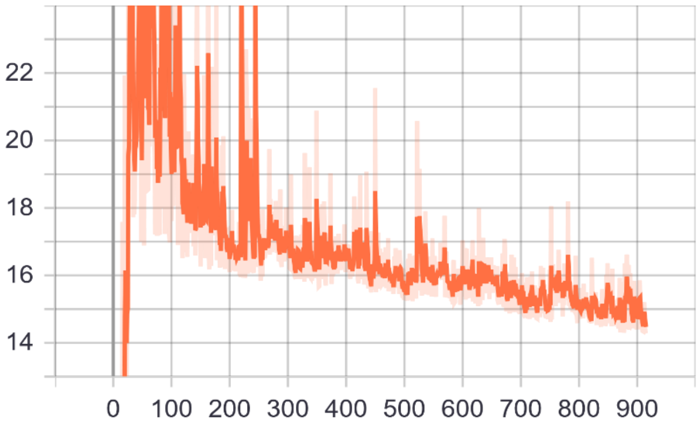


Figure 65Normalized Loss from model ResNet. Visualization from tensorboard.

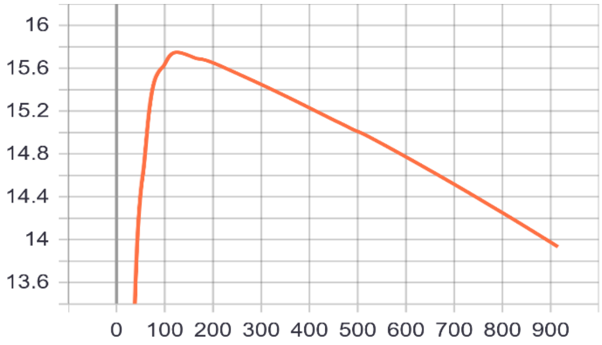


Figure 66Regularization Loss from model ResNet. Visualization from tensorboard.

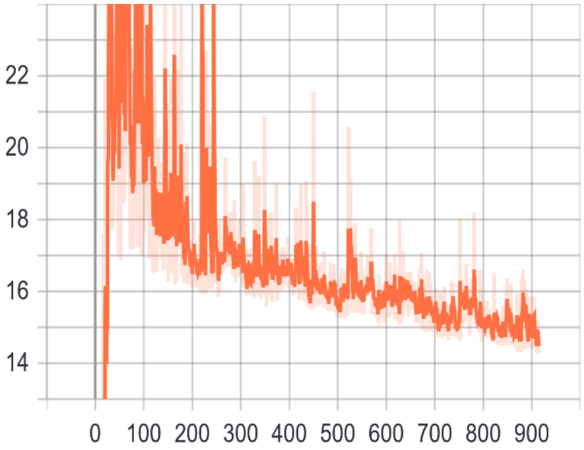


Figure 67Total Loss from model ResNet model during its training using Transfer Learning techniques. Visualization from tensorboard.

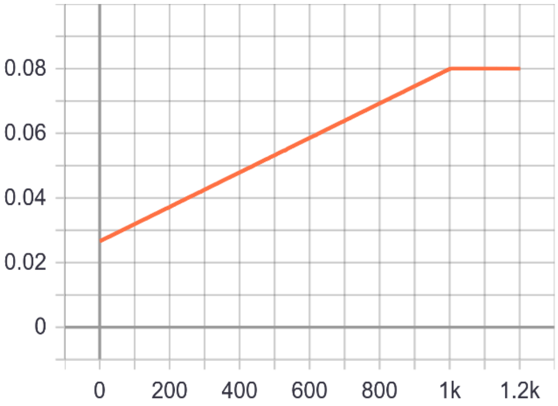


Figure 68Learning Rate for model MobileNet during its training using Transfer Learning techniques. Visualization from tensorboard.

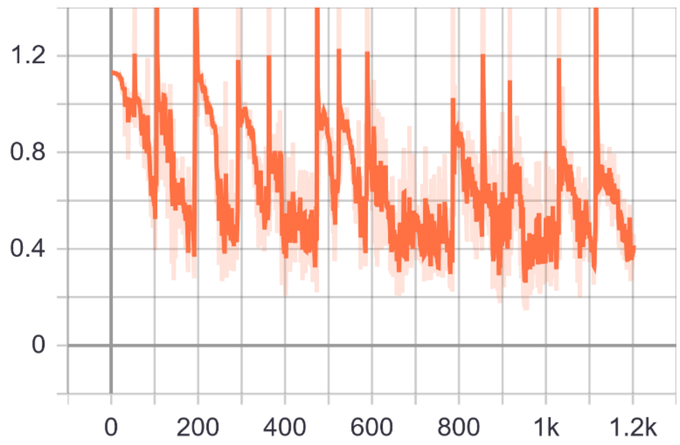


Figure 69Classification Loss from model MobileNet. Visualization from tensorboard.

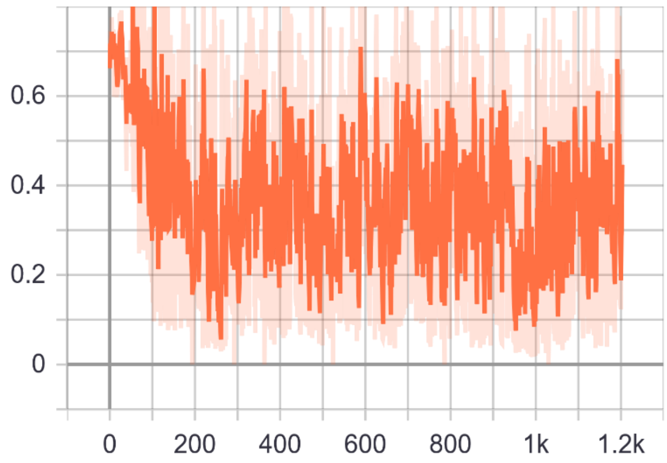


Figure 70Localization Loss from model MobileNet. Visualization from tensorboard.

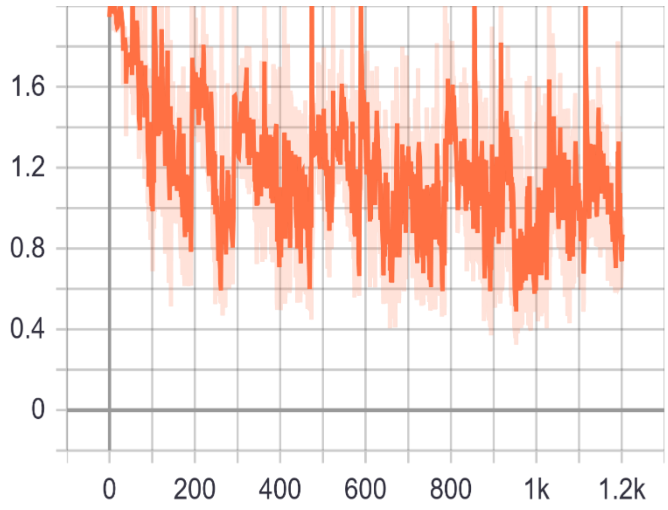


Figure 71Normalized Loss from model MobileNet. Visualization from tensorboard.

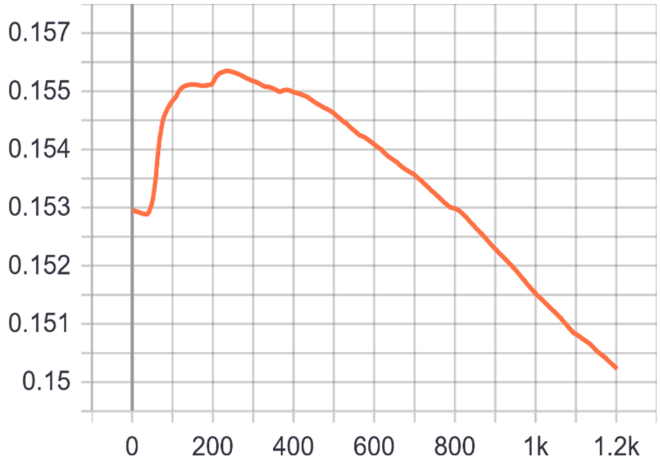


Figure 72Regularization Loss from model MobileNet. Visualization from tensorboard.

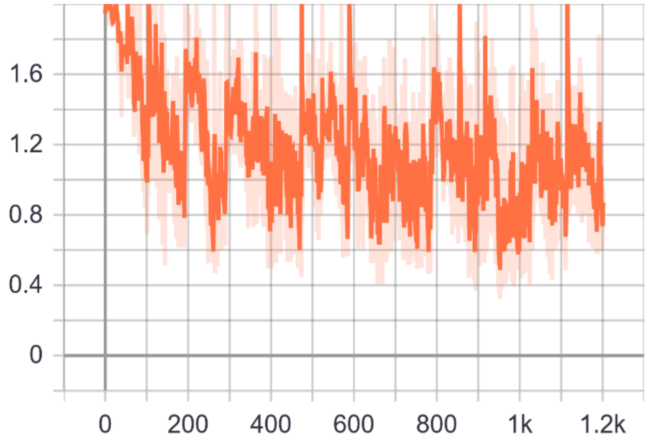


Figure 73Total Loss from model MobileNet. Visualization from tensorboard.

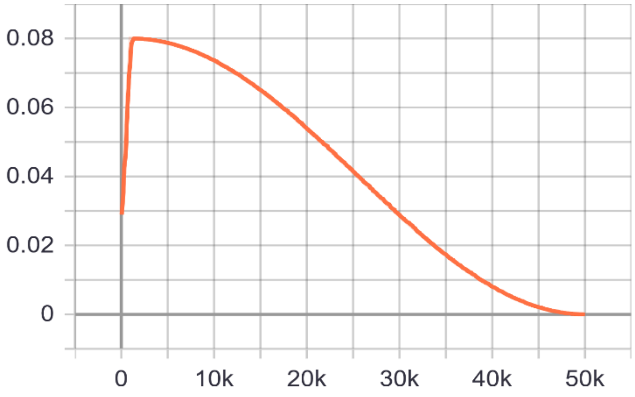


Figure 74New Learning Rate for model MobileNet using 50k steps instead of 25k. Visualization from tensorboard.

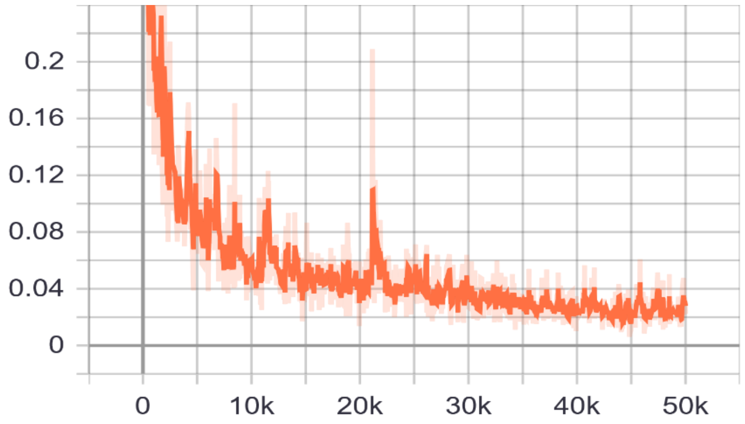


Figure 75New Classification Loss from model MobileNet using 50k Step size instead of 25k. Visualization from tensorboard.

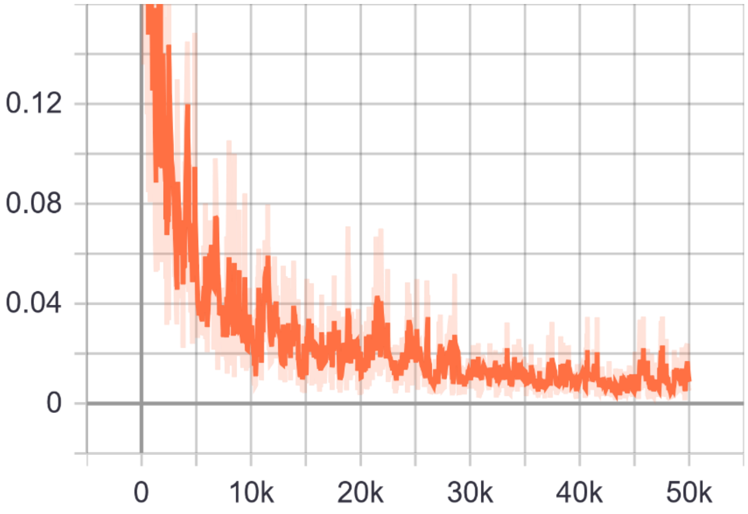


Figure 76New Localization Loss from model MobileNet using 50k steps instead of 25k. Visualization from tensorboard.

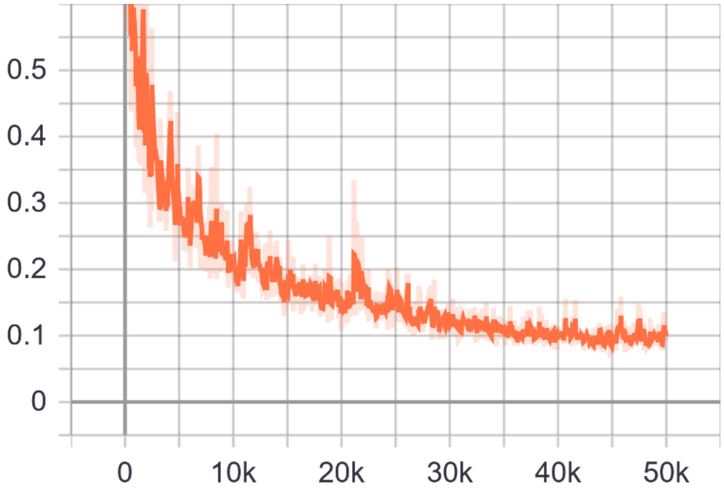


Figure 77New Normalized Loss from model MobileNet. Visualization from tensorboard.

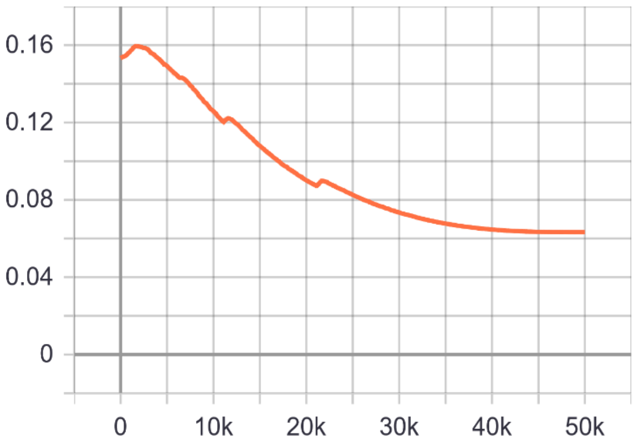


Figure 78New Regularization Loss from model MobileNet. Visualization from tensorboard.

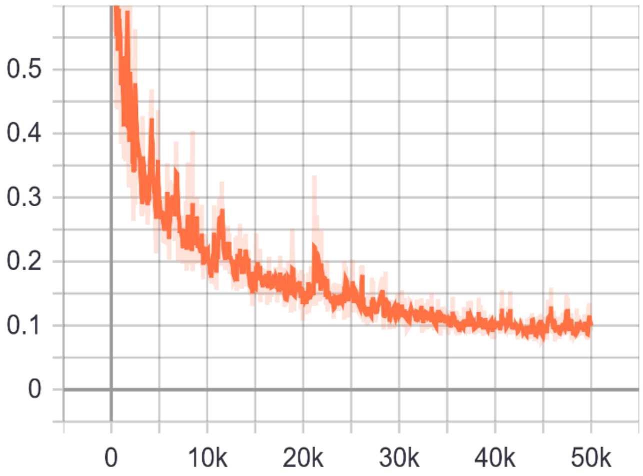


Figure 79New Total Loss from model MobileNet using 50k Step size for training compared with 25k steps. Visualization from tensorboard.