Coventry University

Faculty of Engineering, Environment and Computing

School of Computing, Electronics and Mathematics

DEVELOPING CONVOLUTIONAL NEURAL NETWORKS FOR USE IN DETECTING RAILWAY MAINTENANCE ISSUES

Submitted in partial fulfilment of the requirements for the Degree of Master of Computing: Data Science and Computational Intelligence

Academic Year:
2017 - 2018

Author : Matthew Sumner

Student ID : 8240792

Supervisors: Mauro Innocente, Magesh Nagarajan

Declaration of Originality

Declaration of Originality

This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet, etc) has been acknowledged within the main report to an entry in the References list.

I agree that an electronic copy of this report may be stored and used for the purposes of plagiarism prevention and detection.

I understand that cheating and plagiarism constitute a breach of University Regulations and will be dealt with accordingly.

Copyright

Signed: Date: 20/08/2018

Abstract

In recent years, growing research in Convolutional Neural Networks (CNNs) alongside advances in computing power has led to breakthrough results in computer vision tasks. This paper aims to explore the potential for CNNs to be implemented for use in automated maintenance issue detection in railways. By improving computer vision systems, it may be possible to increase rates of fault detection and reduce the number of man hours required to inspect railways, leading to increased rail safety and reduced costs; both by reducing the need for manual inspection, and by removing the need for blanket rail maintenance. Improvements to computer vision systems could see not only the improvement of current automated systems, but also see the scope of automation in railway inspection expand into areas previously covered by manual inspection alone. To this end the project has explored both the area of railway maintenance and the most sophisticated CNN architectures in recent years and has applied this knowledge through the supervised training and subsequent evaluation of CNNs to classify different issues in sleeper fasteners from a relevant data set containing images collected from current automatic track inspection systems. The work has led to networks with high levels of classification accuracy being produced, particularly through the methods of transfer learning and the ensembling of multiple networks. Insights into the effectiveness of the different regularisation methods employed to overcome challenges specific to the task have been produced. As well as this, an analysis of the practicality of implementing such systems has also been considered, with suggestions on how different technologies and approaches to CNN architecture design may overcome these challenges. Furthermore, the project has looked forward into how the emerging technology of autonomous drones may be able to increase the scope of image data that can be automatically collected, and how CNNs could be used in conjunction with drones in order to detect a wider breadth of maintenance issues.

Contents

Declaration of Originality 1
Abstract 2
Additional Materials on the shared OneDrive 5
Acknowledgements 6
1 Introduction 7
1.1 Overview 7
1.2 Background 8
1.3 Objectives 8
2 Literature Review 9
2.1 Introduction 9
2.2 Image recognition systems and Convolutional Neural Networks 10
2.3 Techniques to overcome data specific issues 16
2.4 Railway Maintenance 18
2.5 Concluding Thoughts 19
3 Methodology 20
3.1 Introduction 20
3.2 Understanding the rail maintenance industry 20
3.3 Gather and label a relevant data set for supervised training 20
3.4 Developing knowledge of CNN technology 21
3.5 Design and evaluation of networks 21
4 Requirements and Analysis 22
4.1 Requirements 22
4.2 Analysis 22
5 Design and evaluation of Convolutional Neural Networks 23
5.1 Data Set 23
5.2 Development Environment 24
5.3 Training 25
5.4 Performance Metrics 26
5.5 Evaluation of regularisation methods 27
5.5.1 Network Design A 27
5.5.2 Network Design B 30
5.5.3 Comparison of Network A and B 32

5.6 Transfer Learning 33
5.6.1 VGG-16 33
5.6.2 Inception_V3 35
5.6.3 ResNet-50 36
5.7 Ensemble 37
6 Implementation into GUI program 38
7 Testing 40
8 Project Management 41
8.1 Project Schedule 41
8.2 Risk Management 42
9 Critical Appraisal 43
10 Conclusions 46
10.1 Achievements 46
10.2 Future Work 47
11 Personal Reflections 49
12 Bibliography 50
Appendix A – Project Proposal 53
Appendix B – Meeting Records 56
Appendix C – Presentation Slides 65
Appendix D– Certificate of Ethics Approval 76

Additional Materials on the shared OneDrive

The following files relevant to the project have been added to the OneDrive folder shared with the project supervisors.

- GUI application and required system files
- Python programs used to train networks throughout project
- Outputs of Neural Networks training in Word Documents
- Data Set

GUI application and Network Training programs require the following prerequisites to run

Python 3.5

Tensorflow library for Python

Keras Library for Python

Matplotlib Library for Python

Numpy Library for Python

SciKitLearn Library for Python

Acknowledgements

I would like to acknowledge my project supervisors, both Mauro Innocente and Magesh Nagarajan, whose support, knowledge and enthusiasm has helped guide the project.

I would like to acknowledge Sam Pashley and Tim Flower of Network Rail, who showed great interest in the project in its initial stages and were able to arrange the invaluable connection with OmniVision.

I would like to acknowledge Trevor Rudd and Stephen Tait of Omnivision, Stephen in particular for his time meeting with me and discussing their current systems and current Deep Learning project and his arranging for the dataset to be supplied for the project. Without Stephen’s insights and the data provided, the work completed in this project would not have been possible.

I would like to acknowledge Peter Lacombe of Derby University, who’s reference was invaluable to me securing my place on the MSc course, and has help lay the foundations to my future in Data Science research.

Finally, I would like to acknowledge my family and friends who have supported me throughout the last year of my studies.

1 Introduction

1.1 Overview

The following paper explores the potential for implementing Convolutional Neural Networks (CNNs) for the automated detection of maintenance issues in railways. The paper begins by introducing the project, stating the motivations for the implementation of such systems, citing recent developments in rail maintenance procedures and current deep learning research occurring in the industry, as well as enabling technologies that have given rise to current automated track inspection. This leads on to a set of stated objectives based on the current state of the industry and how the research in this project may lead to improvements. The paper goes on to a literature review, exploring different computer vision systems, in particular the cutting-edge architectures produced through CNN research, as well as methods used for overcoming issues faced in specific data set problems. The literature review also covers various aspects of the UK rail network and current maintenance procedures, exploring the current systems in place, their limitations and areas where automation could be further expanded. Concluding thoughts are then drawn as to where the implementation of CNNs may be best placed to enhance these maintenance procedures. This research is then carried forward into training and evaluating CNNs to classify the samples in the data set gathered for the project, analysing different regularisation methods and carrying those demonstrating the ability to enhance network performance forward into more complex architectures, leading to higher levels of accuracy. Following this, the paper describes the implementation of the best performing networks into a GUI application that can showcase the potential for the technology. A critical appraisal is made stating the strengths as well as drawbacks to the projects work, discussing the different networks trained throughout the project, assessing both the classification accuracy of the different networks, the scope of the classifications, and the practicality of their implementation into industry. The achievements of the project are stated, and a section on future work discusses how the project could be expanded, both by enhancing the networks built in this project, and how other emerging technologies could lead to a wider range of maintenance issues being automatically detected with the implementation of CNNs. Finally, a section for personal reflection discusses the personal knowledge and skills gained by carrying out the project.

1.2 Background

The use of computer vision has seen a huge growth in the last twenty years, seeing systems being implemented in a wide range of products and industrial applications, such as augmented reality, navigation systems for autonomous vehicles, and automated inspection in manufacturing systems. In more recent years, Convolutional Neural Networks (CNNs) have moved to the forefront in computer vision research, enabling breakthrough results in increasingly complex image recognition tasks (Gershgorn, 2017).

UK rail maintenance groups have been using computer vision systems along with various other sensor apparatus to help automate parts of the rail maintenance process since 2003, which saw the introduction of the New Measurement Train (The Railway Magazine, 2003). Currently Network Rail and OmniVision are working in partnership to explore the use of deep learning to increase the effectiveness of these computer vision systems. It may be that with increased abilities in computer vision, the rail maintenance industry may be able to gain a huge wealth of benefits, with increased accuracy of detection in current systems potentially leading to further improvements in process efficiency, as well as increased safety for both passengers and maintenance personnel. It may also be possible to extend the scope of computer vision in the field, by use of autonomous drones that could capture images of track from a wider range of angles, allowing for a greater variety of issues to be automatically detected.

1.3 Objectives

The project aims to explore the current systems used in rail maintenance in the UK, evaluating current processes, capabilities in automated detection and areas where improved computer vision may aid the sector. Then, by applying design principles seen in the recent developments in CNNs, building and training networks that are able to accurately classify maintenance issues found in railways from images alone.

To this end, the following distinct objectives have been formed:

- Identify common maintenance issues in railways
- Identify areas of maintenance process that may benefit from improved computer vision
- Evaluate the performance of numerous CNN architectures and regularisation methods in classifying a focused area of maintenance issues
- Implement networks into working GUI program, used to demonstrate the capabilities of the networks built and to serve as proof of concept for the potential of such technologies
- Evaluate the effectiveness of different networks, and the practical considerations for implementing them into industry

2 Literature Review

2.1 Introduction

Computer Vision is the field of research that aims to build and implement computational models that can extract information from images or videos (Ballard & Brown, 1982). The motivation behind such systems comes from the potential to automate tasks that previously were only possible with human vision. Examples of such implementation can be seen in a variety of different applications including facial recognition, automatic inspection, and more recently in the developing field of vehicle automation.

The birth of such research began in the 1960s, where Larry Roberts, regarded as the father of computer vision, discussed the possibility of extracting 3D geometrical information from 2D perspective views of polyhedral shapes in his PhD thesis (Huang, 1996). Whilst initial research began with the belief that accurate systems would be developed and available in a matter of months, the lack of computational power at the time led to little progress in the implementation of ideas surrounding the subject (Papert, 1966)

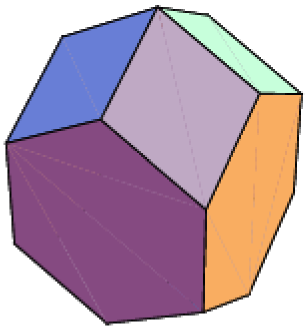

Figure 1: Example of a polyhedral shape, a 3D shape composed of 2D geometric shapes

This literature review has been split into 3 sections. The first aims to cover more recent advances and implementation of computer vision systems, particularly the area of Convolutional Neural Networks (CNNs) for image classification problems. This section forms a justification for the use of CNNs within the project over other computer vision methods and explores their architectural development in recent years. The second section goes on to explore distinct techniques that have been developed in order to overcome practical issues relevant to different dataset problems, particularly those related to the data set used in this project, allowing the acquisition of knowledge and tools to tackle potential problems that may appear during the training of the CNNs in this project.

The final section investigates the processes used as well as the challenges currently faced by railway maintenance groups in order to gain information about current systems in place, where there is room for potential improvement, and what systems may be in place that could be taken advantage of with the implementation of computer vision systems.

2.2 Image recognition systems and Convolutional Neural Networks

Whilst the field of research into computer vision began in the 1960s, it wasn't until the early 2000s with the introduction of the Viola-Jones framework that a robust object detection system was developed that could deliver competitive accuracy in real time (Wang, 2014). The Viola-Jones framework was introduced in the article *"Rapid object detection using a boosted cascade of simple features"* (Viola & Jones, 2001). The framework was designed for the detection of front facing faces. This was done through the creation and implementation of Haar Features, which were used to detect properties that were common to human faces, such as the eye region being darker than the cheeks, and the nose bridge being brighter than the eyes (Viola & Jones, 2001).

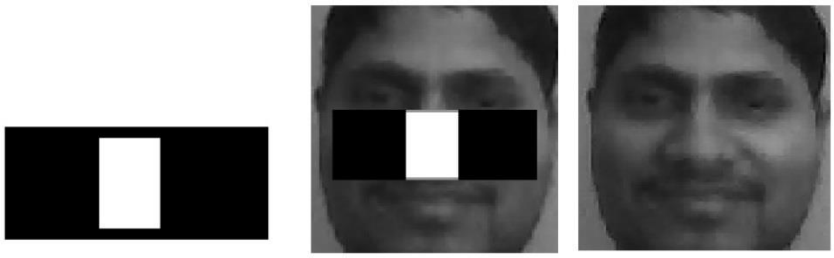

Figure 2: Example of a Haar feature used to detect eyes and bridge of nose.

By computing the difference in pixel values between the white and black areas of the Haar Feature space in different areas of the image it is possible to detect the presence of such features, and by the composition of these different features' location and size in an image the model is able to predict whether or not a face is present in the image.

Whilst the Viola-Jones Framework offered a large step forward for practical implementation of image detection models, the system's use of Haar Features directly relating to the features of human faces makes the model's ability to take on new image recognition tasks limited and time consuming, with new features needing to be manually identified and implemented in order to take on new tasks. Due to the required establishment of pre-defined features in order for the framework to be modified for new tasks, it seems unfeasible for such a framework to be implemented for the task of detecting rail maintenance issues. Such modification of the framework would likely require large teams of specialists in both rail maintenance issues and computer vision, and with such a wide range of different maintenance issues in comparison to the single task of detecting faces, it would be extremely time consuming to produce.

Convolutional Neural Networks (CNNs), by contrast, have the ability to detect, store and then search for features, associating different features and their relevance to multiple classes during training. The first CNN implemented into a real-world problem was the LeNet-5, developed by Yann Lecun in 1998, designed to classify handwritten digits from 32x32 pixel images (Lecun, et al., 1998). The network was implemented by a number of banks, which used the framework to read hand-written cheques, and is seen as the first industry implementation of CNNs for image recognition tasks. Whilst the potential for CNN architecture had been made apparent, a lack of computational power at the time removed the feasibility for such models to be used for more complex, higher resolution images.

It wasn't until the paper "*ImageNet Classification with Deep Convolutional Neural Networks"* (Krizhevsky, et al., 2012) was released that CNNs saw widespread adoption in more complex image recognition tasks (Gershgorn, 2017). The paper describes the research and design of the AlexNet Convolutional Neural Network, which was built to take on the ImageNet Large Scale Vision Recognition Problem (ILSVRC) in 2012. The network consisted of 5 convolutional layers, applying max pooling to the 1st, 2nd and 5th convolutional layer, plus 2 fully connected layers and a final SoftMax activated output layer used to produce the final classification (Krizhevsky, et al., 2012). The network also implemented the recently developed Dropout regularisation method, whereby at each epoch of training, a number of neurons are probabilistically removed from each layer. At the time, it had become apparent that combining networks into an ensemble, whereby multiple neural networks are trained and then implemented together for the same task, would increase the overall accuracy of a system; however computationally it was prohibitively expensive. Dropout aimed to gain this advantage without the increased expense and has proven proficient in reducing over-fitting and therefore testing accuracy by preventing the co-adaptation of neurons in a network (Srivastava, et al., 2014). As a neuron is unable to rely on another at each epoch of training, the network is forced to learn more robust features in order to create its final classification. The AlexNet model came first in the ILSVRC 2012 with a ground-breaking 15.3% top-5 error rate, more than a 10% improvement over the second-placed entry. This result led to a huge shift of research into the networks, with CNNs implemented to win the competition in all subsequent years. The authors noted that the depth of the network was paramount to the network's accuracy, with the removal of a single layer reducing the network's accuracy up to 2% (Krizhevsky, et al., 2012).

In following years, CNNs designed to take on the competition increased in depth, and new novel designs were introduced in order to tackle the problem. The paper *"Going Deeper with Convolutions"* (Szegedy, et al., 2015) introduced the GoogLeNet CNN. The authors introduced the inception module, which was used extensively throughout the deep GoogLeNet architecture. The inception module was noted as the first time that CNN research had extended a network design beyond adding additional layers and altering the perceptive fields. It was also the first model that introduced the idea that CNN layers didn't always need to be stacked sequentially (Deshpande, 2016).

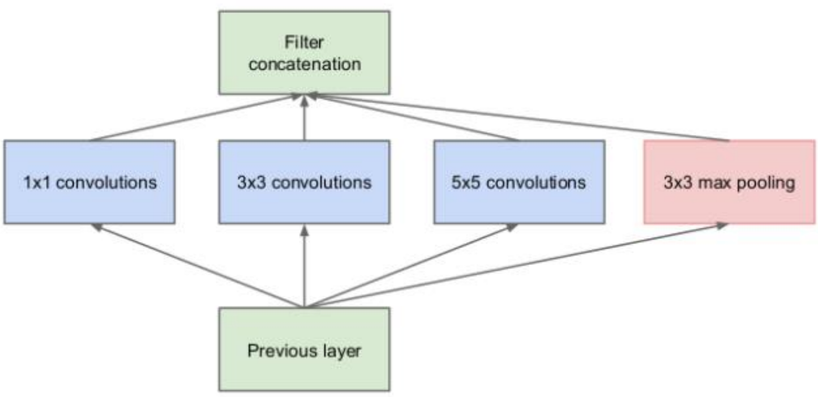

Figure 3: Layout of inception module

The inception module was introduced to find a solution to the on-going debate regarding suitable receptive field sizes for convolutional layers. Receptive fields throughout research had ranged from 1x1 to 11x11 and beyond, however no concrete findings have led to formal guidelines being formed regarding how the size of perceptive fields throughout networks are best suited for different tasks. Rather than having a set perceptive field, the inception module simultaneously implements convolutional layers with 1x1, 3x3, and 5x5 perceptive fields, as well as a max pooling layer simultaneously, outputting a concatenation of these individual layers to carry forward into the subsequent layers of the network (Szegedy, et al., 2015).

The team behind GoogLeNet later released a further paper *"Rethinking the Inception Architecture for Computer Vision"* (Szegedy, et al., 2016). In it they noted some of the primary design principles that led to the creation of the GoogleNet network, and made further improvements to the network, producing the InceptionV3 network architecture, which saw a further 3% decrease in the top-5 error rate when compared to its predecessor. The authors noted the following design principles they feel should be followed when designing an accurate and computationally efficient neural network:

- Avoid bottlenecks with extreme compression – Whilst the authors note that to reduce the network's parameters to a manageable amount, pooling techniques are necessary, these pooling techniques should reduce the size of the input gradually so as not to lose valuable information from the input image. (Szegedy, et al., 2016)
- Distribute the computational budget between height and width of the network – When extending a network's number of parameters, proportional extension of the depth of the network (number of layers) and width of the network (number of parameters at each layer) will lead to better accuracy over focusing on the extension of one of these variables. (Szegedy, et al., 2016)

Whilst these design principles were formed through the extensive training of extremely large neural networks, the authors note that they are relevant to the design of CNN's period, and should be followed to create models that are both accurate and computationally efficient.

Following such design principles as stated in the paper may offer aid in making good design choices within the network, which are imperative to maximising the accuracy of models built when dealing with the limited computational resources available.

Whilst GoogLeNet was the winning entry in the ILSVRC2014, the second-placed entry is also highly regarded. The Paper *"Very Deep Convolutional Networks for Large-Scale Image Recognition"* (Simonyan & Zisserman, 2015) presents the different VGG network architectures and analyses their results. The architecture of VGG networks is characterised by the stacking together of multiple convolutional layers with the same 3x3 perceptive field before max pooling layers. The reasoning behind such design was the authors' assertion that the combination of two convolutional layers of 3x3 perceptive field size have a combined effective perceptive field of 5x5, as the second convolutional layer is searching over a map created by the reduction of 3x3 pixel windows into 1x1 representations. This methodology is also extended to include three simultaneous convolutional layers, creating an effective perceptive field of 7x7 (Simonyan & Zisserman, 2015).

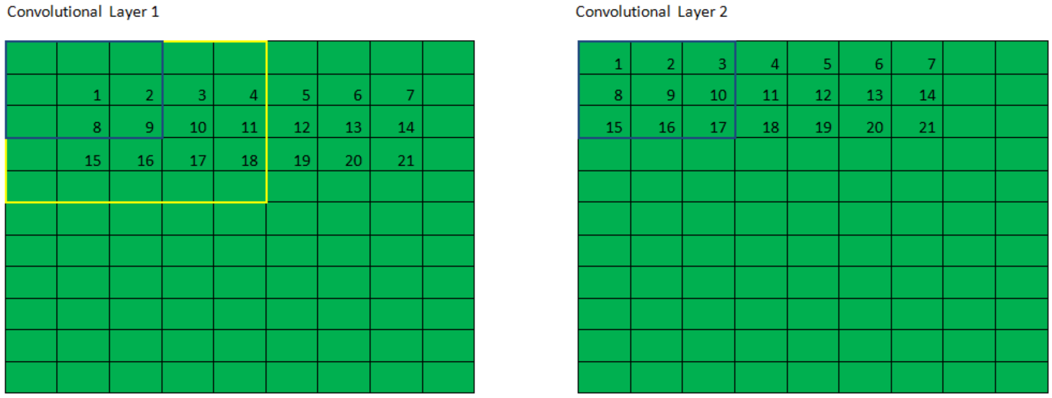

Figure 4: comparison showing how 2 3x3 perceptive fields (blue) can have the same effect as a single 5x5 perceptive field (yellow). Note that each pixel captured in the 5x5 filter is also captured in the 3x3 perceptive field in the second layer.

Although not winners of the competition, the authors of the paper were praised for their unique way of dealing with the issue of suitable perceptive field size, as well as the relative simplicity of the network design. In fact, in the modification of the GoogLeNet architecture seen in InceptionV3, it included the use of stacking two convolutional layers with 3x3 perceptive field sizes in place of 5x5 perceptive fields previously implemented in the network's inception modules (Szegedy, et al., 2016). Unlike GoogLeNet, the VGG network represents a more standard CNN implementation. Due to their relative architectural simplicity, the VGG networks such as VGG-16 and VGG-19 have become popular in transfer learning, whereby pre-trained neural networks are modified and fine-tuned to tackle new tasks. This simplicity makes implementing and fine-tuning the various layers easier when compared to more complex networks. However, whilst the architecture is simple to comprehend in comparison to others, the large number of parameters compared to GoogLeNet leads to considerably longer training times, which may see the full network architectures being unfeasible for use in this project, at least if trained as a naïve model (using randomly generated weights rather than those gained from the ImageNet training). Whilst full implementation of VGG-16 or VGG-19 may be unfeasible for extended training, the design principles behind such networks are simpler to implement compared to a network employing more novel approaches and they have proven to produce accurate models. As such they may help form the projects initial network designs, where scaled down versions of the networks may allow for quicker training and reasonable accuracy.

Continuing the trend of increasingly deep neural networks for image classification, the winning CNN of the ILSVRC 2015, ResNet was presented by a team from Microsoft Research Asia in their paper *"Deep Residual Learning for Image Recognition"* (He, et al., 2016). The largest of the network architectures described in the paper boasted an incredible 152 layers, resulting in training time of up to three weeks, even with machines running 8 state of the art GPUs at the team's disposal. The best performing ResNet network achieved a 5.71% top 5 error rate (He, et al., 2016), beating all other entries in the image classification competition. Extending this, the authors created an ensemble of 6 networks of varying depth, whereby the mean values of each output layer is taken, which are then used to make the final classification prediction. This method further reduced the top 5 error rate to 3.57%, a remarkable result owing to the fact that the error rate of humans ranges from 5-10% dependent on the individual's expertise in the subject matter of the images, showing that the network could actually outperform the vast majority of humans in the task (Deshpande, 2016). As well as its depth, ResNet's architecture was characterised by its implementation of newly developed residual blocks. These blocks allow the input of a layer to skip connections, allowing the input to be fed to layers deeper in the neural network. It has been shown that with increasingly deep neural networks, layers later in the network can struggle to learn useful features, which in turn leads to a decrease in accuracy compared to more shallow networks. By allowing earlier layers to take shortcuts into later layers, it mitigates the chance of extremely deep networks losing their ability to form the identity function (Ng, 2018).

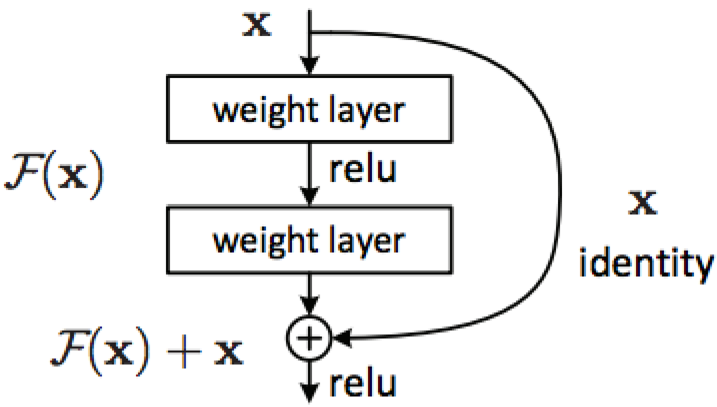

Figure 5: Diagram illustrating functionality of residual block

Whilst this implementation allowed for incredible results by reducing the issue of decreased accuracy in very deep neural networks, the authors noted that even with the residual block, naïvely adding huge numbers of layers could still be detrimental to the accuracy of the network. They discovered this by training a residual neural network of 1202 layers, which led to a higher error rate than the shallower networks in the paper. They believed that with such a huge number of parameters in the network, it is likely that it had caused overfitting to the training data, reducing its ability to generalise at the testing stage (He, et al., 2016).

Through the training and analysis of different neural network architectures throughout the project, it may appear that certain designs may show varying classification abilities in different classes. In such scenarios the ensembling of different networks may allow for further increases of accuracy. With the most time-consuming process of constructing CNNs being training, the ensembling of networks may offer a time efficient way of increasing the accuracy of the overall finished product.

2.3 Techniques to overcome data specific issues

Whilst the networks discussed in the previous section show the cutting edge of CNN capabilities and offer valuable information with regard to the design of CNNs, they also took advantage of large time scales, huge computational power and vast data set sizes. The data set, computational power and time available to this project are significantly smaller. As such training large complex models from scratch maybe unfeasible. The paper *"Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?"* (Tajbakhsh, et al., 2017) identified the practical challenges faced when trying to build and train a CNN from scratch when limited data and computational power is available. They explored the potential of transfer learning, where previously trained and proven neural networks can be repurposed and fine-tuned in order to tackle new problems. The paper took the AlexNet CNN architecture and altered its output layer in order to detect the presence of polyps (an abnormal growth of tissue projecting from a mucous membrane) in images taken from colonoscopy videos. They compared the accuracy of the network through different levels of fine tuning, where different layers have their weights frozen from previous training whilst others are trained on the new data set. Throughout the study they found that fine tuning all layers in the network produced a greater accuracy of results compared to training a Naïve AlexNet model, and even moderately fine tuning the final fully connected layers of the model yielded impressive results, whilst significantly reducing the number of samples and time needed to train the network to achieve high accuracy compared to training from scratch (Tajbakhsh, et al., 2017).

On the matter of data set size, data sets can also be extended artificially with the use of Data Augmentation techniques. Data Augmentation occurs where new samples are created in the set by performing modifications to the pre-existing data samples. In the case of image classification this involves the modification of an image whilst still retaining its defining features. The paper *"The Effectiveness of Data Augmentation in Image Classification using Deep Learning"* (Perez & Wang, 2017) explored the use of data augmentation and compared the effectiveness of different techniques in improving the accuracy of the network. Throughout the paper the authors used data augmentation to double the size of the data sets used, by creating an augmented image from each of the original samples. They compared traditional augmentation techniques such as zooming, rotating, and flipping with more complex techniques, such as the use of Generative Adversarial Networks in order to create distinct image 'styles' that could be used to extend the size of the data set (Perez & Wang, 2017). Whilst different augmentation methods produced more accurate models dependent on the task, in all cases, networks trained with the implementation of data augmentation consistently outperformed those that didn't, with up to a 7% increase in network accuracy observed in certain problems.

In addition to dealing with a small data set, the number of examples of different classes may vary significantly. Images of sections of rail in good working order are likely to be more common than those with issues, causing there to be an imbalance of class samples in the data set. Imbalanced class samples in data sets can often lead to unwanted effects on a network's ability to correctly classify an image. The presence of class imbalance often leads to networks becoming more biased to classifications with a higher number of samples (Buda, et al., 2017). In the paper *"A systematic study of the class imbalance problem in convolutional neural networks"* the authors explore a number of different methods that aim to mitigate issues caused by class imbalance, particularly in the field of image classification (Buda, et al., 2017). The paper explored several Data Level and Classifier level methods, including:

Data Level

Under Sampling: Reducing the number of samples of classes with more examples, so that classes have a more even number of samples in the data set.

Over Sampling: Increasing the number of images in the smaller classes by adding duplicate copies of pre-existing images, leading each class to have more balanced number of samples.

Classification Level

Thresholding: Adjusting the output of the final classification layer at testing, by dividing each output by its Bayesian posteriori probability, defined by

$$y_i(x) = p(i|x) = \frac{p(x|i) \times p(i)}{p(x)}$$

Cost Sensitive Learning: Adjusting the cost function of the network to assign a different cost for misclassifying different classes.

The paper defined Class imbalance as two separate type. The first type (type 1) occurs when each majority class and each minority class hold similar numbers of examples, but a significant disparity in the number of examples between majority and minority classes are present. Type 2 occurs when there is a significant disparity in the number of samples in each class (Buda, et al., 2017). The paper showed that with increasing class imbalance of both types, networks trained on such data sets were likely to lose accuracy and become biased to majority classes, and this loss of accuracy was increased greatly with increasingly complex image recognition tasks (Buda, et al., 2017). It was concluded that at the data level, oversampling consistently yielded the greatest improvements to accuracy, being most effective when employed to make each number of class samples equal. Thresholding was the most effective classification level method for mitigating the class imbalance issue, with the best results being produced by a combination of oversampling and thresholding.

2.4 Railway Maintenance

With over 1.7 billion passenger journeys made in 2017, Britain's rail network is an integral service enabling passengers to commute to work, visit family and travel the nation. Of all passenger journeys, 69% are made in the London and South East areas (excluding underground), where increasingly high road traffic in the area makes the rail network a crucial part of the City's infrastructure (Department For Transport, 2017). The use of rail transport is increasing year on year, with a 20% increase in Q1 passenger journeys seen between 2012 and 2017. With this trend of increasing rail use, the importance to develop more efficient rail maintenance systems to tackle the ever-growing maintenance requirements of British Rail networks seems self-evident. Indeed, in a 2017 Rail Industry Survey, operational & maintenance costs were identified as the most prominent issue currently faced by the industry (Nomad Digital, 2017).

One of the most significant developments in UK rail maintenance in recent years has been the introduction of the New Measurement Train (NMT) in 2003. Built as a direct response to the concern over gauge corner cracking following the Hatfield rail crash in 2000 (Rail Magazine, 2003), the converted passenger train is equipped with a range of equipment ranging from under and over carriage cameras, gyroscopes, accelerometers and lasers. The underside cameras in particular capture image data used to detect issues in sleeper fasteners by means of the Plain Line Pattern recognition system, which compares the images collected to images of rail in perfect condition, in order to detect the presence of issues (Network Rail, 2017).

The NMT and the fleet of 4 other similar trains periodically traverse the nation's rail lines, collecting images along with sensor readings measuring the locomotives' movement over the line, combining this data to detect various issues with the line. As well as improving the detection rates of a wide variety of maintenance issues, the NMT fleet has improved the safety of the rail maintenance field by reducing the amount of time staff are required to traverse the rail tracks, which has also led to a more efficient maintenance process. Furthermore, costs have been reduced through targeted maintenance, whereby only specific parts of the rail line that need to be replaced are dealt with, as opposed to costly blanket renewals of large sections of railway (Denham, 2012).

Figure 6: The New Measurement Train

Whilst the fleet of measurement trains on the national rails has helped increase the detection rate of faults whilst reducing the number of man hours needed for maintenance processes, there are some apparent drawbacks. The trains themselves run daily, and the running and maintenance of the trains is costly. It may be that with increased accuracy of computer vision systems deployed, several maintenance issues could be detected by images alone. With such improvements, implementation of alternative image capturing methods such as autonomous drones may be able to reduce the operational time requirements of the measurement vehicles, saving operational costs as well as the environmental impact of running the trains. Currently the NMTs are only able to provide analysis for plain rail lines and cannot analyse the conditions of switches and crossings adequately (Denham, 2012). With the improved computer vision systems and the increased number of potential viewing angles enabled by the drones, these more complex sections of the rail network may also benefit from further improvements in the maintenance process.

As well as the limitations on the sections of rail that can be analysed by the NMTs, the current system is also prone to giving a large number of false positives for the detection of issues (roughly a third of detected issues are legitimate) (Tait, 2018). Currently each potential issue detected by the NMT is checked manually in one of Omnivision's processing offices. With such a high rate of false positives, large numbers of staff are required to validate the identified issues, and create the reports outlining these issues to be passed on to the relevant local maintenance teams. As well as increasing the cost, increased manual inspection leads the system to be more prone to human error, where factors such as tiredness and boredom may lead to mistakes being made in the validation process. Again, increasing the systems' computer vision capability through CNN's may well be able to reduce these false positives and increase the efficiency and effectiveness of the system.

2.5 Concluding Thoughts

Throughout the literature review it has been demonstrated that whilst other image recognition systems have proved proficient in specific tasks, the use of Convolutional Neural Networks has proved to be extremely effective in adapting to a variety of different image classification problems, and has seen widespread adoption in both industry and academic research. Insights have been gained into the design choices and regularisation methods that have resulted in increasingly accurate models, many of which can be directly implemented into the design choices of the project.

Furthermore, through the study of different surveys, reports and articles about the railway maintenance sector and its processes, the justification for such research aiming to improve the systems is evident. Through increasing the image classification capabilities of such systems by means of CNNs, capabilities of automatic track inspection can be improved and broadened, allowing for safer and more cost-efficient systems.

3 Methodology

3.1 Introduction

This section details the methodology used throughout the project. In its various sections it covers the approaches taken, along with justification as to how these actions will lead to meeting the projects stated objectives.

3.2 Understanding the rail maintenance industry

To form a project that is relevant to the field of railway maintenance, research into the current processes, technology and challenges faced by the industry must first be established. To this end, a section of the literature review will be dedicated to this area, exploring various articles, surveys and maintenance journals from the rail industry. In doing so it will be possible to identify where there is a need for further improvements in the processes, and how the use of improved computer vision by means of CNNs can be best placed in the project.

As well as the literature review, through making contact with Network Rail and their computer vision partners OmniVision, further information will be gained into the current systems in place, their future projects for the system and the areas for improvement that motivates such developments. In addition to extending the information gathered for the project, through face to face meetings a data set may be procured, with a more focused research area formed for the project.

3.3 Gather and label a relevant data set for supervised training

For CNNs to be trained, a data set must be gathered containing adequate samples of relevant maintenance issues. This will be sourced either by reaching out to Rail maintenance groups, or via online data repositories. Once gathered, the samples in the data set must be labelled with the relevant classification, and split into training, validation and test sets, allowing for the supervised training of the CNNs, and analysis of the subsequently trained networks.

3.4 Developing knowledge of CNN technology

As the project has a limited amount of computing power available, it is imperative that intelligent design choices are made in order to maximise the accuracy of the networks with the resources available. Research will be undertaken into the most prominent CNN architectures that have been designed in recent years, as well as a more moderate look into alternative computer vision systems. By doing so, the development of network architectures produced by respected research groups will be analysed for their performance, design choices and feasibility for implementation within the project.

In addition to the advances in CNN design, research will be undertaken into methods of dealing with challenges directly related to the design of networks in this project. Different methods for overcoming small and imbalanced data sets will be explored, allowing the acquisition of tools which may help in further improving the accuracy of networks trained throughout the project.

To grasp the concepts relating to CNNs, the Feynman Technique will be employed, allowing for a full understanding of the concepts and how best to implement them into the papers research.

3.5 Design and evaluation of networks

Following design principles gained from research, initial CNN architectures will be built and trained multiple times, employing various regularisation methods in order to evaluate their effectiveness on the given problem. Instances of the network architectures will be trained on the data set produced for the project, as well as the CIFAR-10 dataset, allowing for a more thorough analysis of performance. Due to the large time required to train neural networks employing large numbers of parameters, initial models will consist of a more moderate number of parameters, allowing for faster training. Whilst these networks are unlikely to produce the most accurate results, shorter training will allow for a larger number of networks to be trained and therefore a more thorough evaluation of the effectiveness of various methods ability to increase accuracy.

Following on from the initial networks, increasingly deep networks employing a larger number of parameters will be trained, implementing the most promising methods deduced from initial training, expected to further increase the accuracy of the best networks. Final networks producing the best results will be further evaluated as ensembles to deduce whether further accuracy improvements can be made before implementation into the final program.

4 Requirements and Analysis

4.1 Requirements

The client has requested that a GUI application be delivered that implements CNNs to produce an output classification of fastener maintenance issues based on the input image with high accuracy (above 80%). The application should allow users to select and classify images within the GUI, returning the classification in a format understandable by rail maintenance staff with no understanding of the workings of CNNs, with the ability to more thoroughly analyse the networks output if necessary.

4.2 Analysis

The functionality requirements of the GUI application are reasonably simple; however, it is the accuracy of the CNNs employed in the application that are fundamental to the product being successful. With the expectation of highly accurate results, the primary focus in design will be towards different CNN architectures and regularisation methods. The iterative analysis and extension of networks will allow for increases in the final application's ability to correctly classify the different maintenance issues.

5 Design and evaluation of Convolutional Neural Networks

5.1 Data Set

The data set used for the training and testing of networks has been provided courtesy of Omnivision. The initial set consisted of 1,000 images of 2048x6432 resolution, with each image containing roughly 7 sleepers and 14 fasteners, in various conditions.

Due to the prohibitive computational expense of inputting such high-resolution images into a CNN, as well as the presence of different fastener classes in each image, a subset of 400 lower resolution images containing single fasteners has been extracted from the original data set.

Figure 7: Sample image from original dataset

The subset of images gathered from the original dataset has been grouped into 3 distinct classes.

No Issue: The Fastener is in good working order.

Missing: The Fastener is missing from the sleeper

Obscured: The Fastener is obscured by ballast, making it impossible to deduce the condition of the fastener using computer vision.

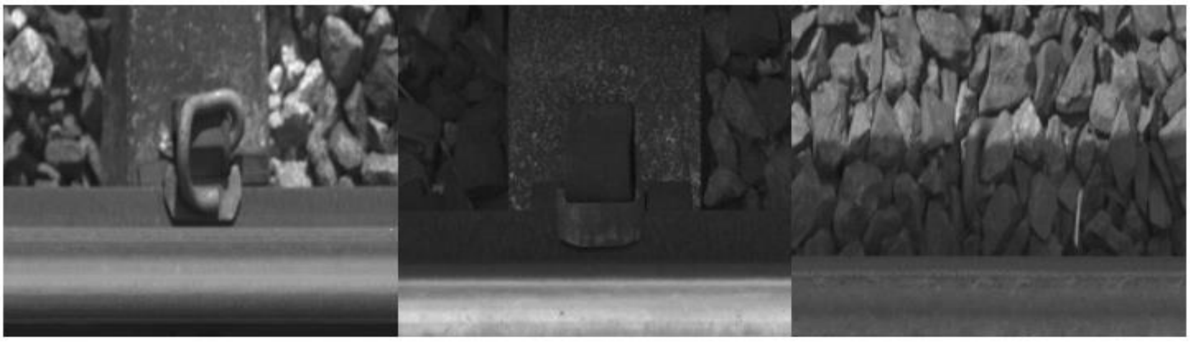

Figure 8:Samples of the data set created from extracting fasteners from original data set. Classes from left to right. No Issue, Missing, Obscured

Other fastener maintenance issues were present in the data set, however due to the scarcity of samples (between 10-20 each) it was not feasible to use these classes for training.

5.2 Development Environment

The networks have been built and trained using the Python 3.5 language and the Spyder IDE. The following libraries have been used to extend the Python language's functionality.

- **Tensorflow**: An open source library used for a range of dataflow programming tasks. The library allows the building of multi-dimensional arrays called tensors, which allow the representation of weight connections between multiple neurons in the layers of a neural network.

- **Keras:** An open source neural network library designed to run on top of tensor flow. It allows for faster implementation of neural network architectures compared to using Tensorflow alone. The library also allows the importing of different neural network architectures and their weights, enabling the implementation of high performance architectures for use in transfer learning

- **Numpy** – An open source scientific computing package that extends the mathematical functionality of python. Numpy has been used to transform the input images into numpy arrays which allow the networks to accept the input image and has also been used to identify the classification output of a network by analysis of the output layer.

- **Scikit-learn** - A Machine Learning Library used for the implementation of a wide variety of machine learning techniques. In this project the package has been used to allow further evaluation of the neural networks performance, allowing the creation of confusion matrices, and the calculation of performance metrics beyond overall accuracy, such as precision and recall, allowing for a more detailed evaluation of network performance.

- **Matplot-lib** – Allows the plotting of graphs, used to visualise network's accuracy throughout training.

The architecture and weights of trained neural networks have been saved in Hierachial Data Format (HDF5). This allows for network's architecture and weights to be saved and loaded mid training, as well as for implementation into the final GUI application.

5.3 Training

The networks throughout the project have been trained by means of the Backpropagation algorithm. The Backpropagation algorithm consists of two phases, the forward pass, where the input samples are presented to the network, and the backward pass, where the size of the error is calculated by means of the loss function, and the weights of neuron connections are adjusted by means of the chosen optimizes. Stochastic Gradient Descent has been used as the optimizer throughout the project, a variation of standard (batch) gradient descent which reduces the computational cost when dealing larger data sets, as it doesn't require the calculation of the cost across all samples each time the optimizer calculates the direction of the gradient (Ng, 2018). By calculating the partial derivatives for each weight in respect to the cost function, the weights are adjusted by the equation:

$$w_{t+1} = w_t - lr\frac{\partial L}{\partial w_t}$$

Where L is the loss function, w is the weight and lr is the learning rate, used to scale the size of the weight changes throughout training.

The Stochastic Gradient Descent optimizer has been further modified to include a momentum coefficient. This gives the optimizer a short-term memory, allowing it to remember the direction of the gradient in previous examples. This allows for faster convergence if the direction of the gradient is the same as the previous iteration and slows convergence if the direction of the gradient has changed. This allows for the escaping local optima and the reduction of oscillations respectively. The equation for updating the weights with the momentum coefficient is described as:

$$w_{t+1} = w_t - lr\frac{\partial L}{\partial w_t} + \alpha[w_t - w_{t-1}]$$

Where α is the momentum coefficient, scaling the effect of momentum in the optimizer.

A learning rate of 0.001 and a momentum coefficient of 0.9 was applied throughout naïve model training, as this seemed to strike a balance between larger oscillations in the cost function output and ensuring weight updates were sufficiently large enough to train efficiently.

5.4 Performance Metrics

The following performance metrics have been used to analyse the performance of the networks trained:

Accuracy: The ratio of correctly classified samples across all classes

$$Accuracy = \frac{True\ positives}{number\ of\ samples}$$

Recall: Ratio of a class's samples that were correctly classified

$$Recall = \frac{True\ positives}{True\ Positives + False\ Negatives}$$

Precision: Ratio of a predicted class that were correct

$$Precision = \frac{True\ positives}{True\ Positives + False\ Positives}$$

F1: Harmonic mean of precision and recall, used in preference to arithmetic mean as it punishes extremely low values of either metric, (e.g. if a network predicts all inputs as the same class, with an even number of class samples, the mean of precision and recall for this class would be $0.\dot{6}$, whereas the harmonic mean would be 0.249). This gives a better representation of the networks classification abilities.

$$F1 = 2\left(\frac{precision \times recall}{precision + recall}\right)$$

Figure 9: Equation for calculating F1 score

By studying the precision and recall as well as the accuracy, it is possible to make a more thorough assessment of a given network's capability across all classes.

5.5 Evaluation of regularisation methods

Initially, CNNs with a moderate number of parameters were built to evaluate various regularisation methods, that may help tackle distinct problems relating to the data set. Whilst these networks were not expected to achieve the accuracy levels laid out in the requirements for the application, the smaller number of parameters that are adjusted throughout training leads to shorter training times. This has allowed for a more thorough analysis into regularisation methods, and therefore greater insight is gained into their effectiveness, allowing these findings to be applied to subsequent, more sophisticated architectures. Methods aimed to mitigate the following data set problems were explored:

Size

The data set obtained for the training and building of networks is limited in size. Typical CNNs trained for image classification have thousands of images for training. By contrast, the data set created for training contains just 400 images, thus the value of exploring techniques such as image augmentation to artificially extend the size of the data set is clear.

Class Imbalance

As well as limitation in the number of samples, the data set contains a much higher number of fastener images with no issue, as fasteners in good working order are much more common than those missing or obscured (the data set is built on a ratio 200:100:100 no issue, missing, obscured). Neural networks trained on data sets exhibiting class imbalance are often prone to exhibiting bias towards majority classes (Buda, et al., 2017). Due to this, methods for mitigating class imbalance identified in previous work will be compared to see their effectiveness on the data set.

5.5.1 Network Design A

The first network design has been designed in line with design choices of the VGG Neural networks (Simonyan & Zisserman, 2015), where multiple convolutional layers are grouped between max pooling layers, creating effective perceptive fields of 5x5 (two stacked convolutional layers) and 7x7 (three convolutional layers) whilst keeping the functionality of 3x3 perceptive fields. In addition to the VGG style design of the network, dropout regularisation methods have been implemented to prevent the overfitting of the network (Krizhevsky, et al., 2012), at a rate of 0.1 following max pooling layers and 0.25 following full connected layers. Whilst the network takes inspiration from VGG-16, the network has fewer layers and parameters, allowing it to be trained multiple times in order to analyse various methods.

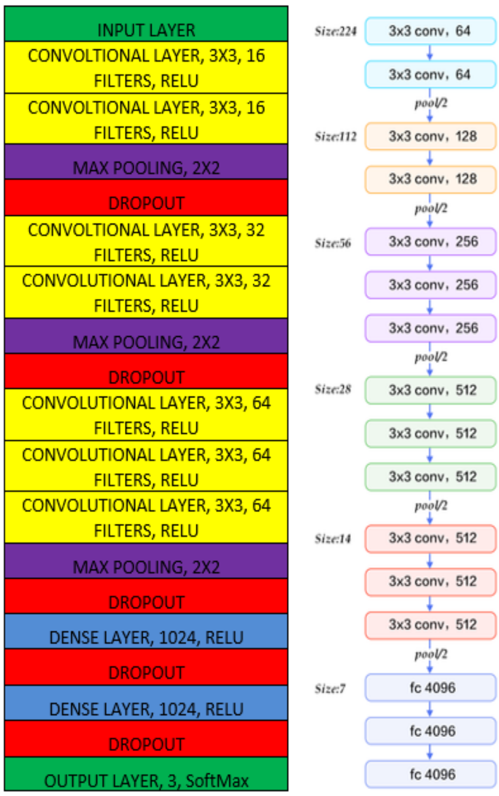

Figure 10: Comparison of Design A and VGG-16

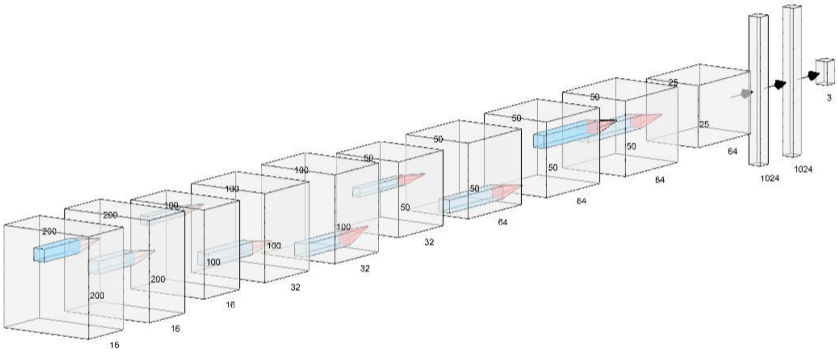

Figure 11: 3D representation of Network A, illustrating the filter sizes and dimensionality reduction throughout the Convolutional and max pooling layers before the fully connected and SoftMax output layers

5.5.1.1 Data Augmentation analysis

Image Augmentation has been shown to improve CNNs accuracy in image classification, particularly when a limited data set is present (Buda, et al., 2017), with a wide variety of different techniques available that have been demonstrated to increase a network's ability to generalise and therefore help avoid the pitfall of overfitting the data set to the training set.

Augmentation has been applied to each image using the following techniques and parameters:

- *Vertical Flip: Flips the image along the x axis, intuitively seems appropriate as fasteners are present on both sides of the rail*
- *Zoom Range (Set 0.3): Randomly zooms into the image, expanding its effective size to between 100-130%*
- *Width Shift (Set 0.1): Randomly shifts the image along the x axis by between 1-10% of total image width*
- *Height Shift (Set 0.1): Randomly shifts the image along the y axis by between 1-10% of total image width.*
- *Rotation Range (Set 90): Randomly rotates the image between 0^{o} and 90^{o}*

By combining these techniques in various combinations, 10 images have been produced from each unique image in the data set, expanding the effective number of samples for training.

Figure 12: Example of output images from applying data augmentation to a single sample

	No Augmentation	Augmentation
Rail Maintenance	27.66%	**36.67%**
CIFAR-10	67.34%	**70.15%**

Figure 13: Test accuracy for network Design A, comparing image augmentation against no augmentation for both Rail Data Set and Cifar-10. Epochs: 50

By studying the accuracy of networks seen in figure 13, it can be seen that accuracy has improved in both image recognition tasks. Whilst the most significant increase in accuracy has been observed in classifying the Rail Maintenance data set, low accuracy scores for both of these networks make it difficult to draw hard conclusions as to its effectiveness. However, the improvement seen in the CIFAR-10 accuracy is encouraging, as both these models have produced reasonably accurate results, especially when considering the larger number of potential classes present in this set.

5.5.1.2 Class Imbalance Methods Analysis

In order to address the issue of class imbalance, the three additional instances of the network have been trained, each applying one of the following techniques:

- Undersampling – Reducing the number of samples of majority classes to create an even ratio of all class samples
- Oversampling – Increasing the number of samples in minority classes by duplicating images to create an even ratio of all class samples
- Thresholding – Normalise the output layer values by dividing SoftMax values by each class's Posterior Probability

No implementation	Undersample	Oversample	Thresholding
36.67%	40%	**46.67%**	43.67%

Figure 14: Accuracy Results for Network Design A, trained multiple times using different methods to mitigate class imbalance. Epochs: 50

By studying figure 14, it can be seen that the methods employed to mitigate the class imbalance in the data set have yielded more accurate results when compared to no implementation. The results align with the findings of (Buda, et al., 2017) who found that Oversampling and Thresholding most frequently achieve the greatest improvement in accuracy when compared to the other methods.

5.5.2 Network Design B

Due to the low accuracy exhibited in all instances of Network A, it seemed prudent to extend the network's architecture, with the expectation that a deeper network employing a larger number of parameters will deliver higher levels of accuracy at testing (Krizhevsky, et al., 2012). The number of filters used at each convolutional layer has been doubled, and an additional full connected layer has been added, resulting in 3 fully connected layers being implemented between the convolutional and output layers. In line with recommendations (Szegedy, et al., 2016) regarding balancing depth and width in a network for computationally efficient networks, the number of neurons in the fully connected layers has been increased by 50%. Dropout rates have initially remained the same. By creating networks that deliver a higher accuracy, stronger conclusions thorough the analysis of methods can be made.

Figure 15: Description of layers in Network B

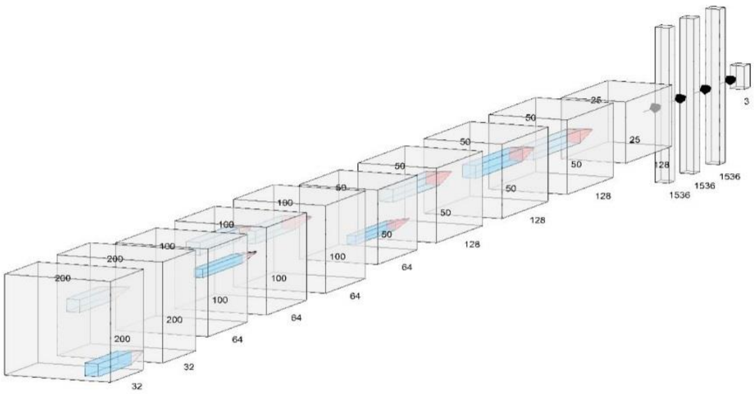

Figure 16: 3D representation of Network B

5.5.2.1 Class Imbalance Methods Analysis

No Implementation	Undersample	Oversample	Thresholding
26.666	40%	**63.33%**	43.33%

Figure 17: Accuracy Results for Network Design B, trained multiple times using different methods to mitigate class imbalance. Epochs:50

Observing the results seen in Figure 17, similar changes in accuracy scores can be observed when comparing the accuracy of Network A's networks. A far more significant increase has been observed with the method of Oversampling compared with previous observations, with the network employing Oversampling returning a 20% more accurate model when compared to the 2nd placed network (employing Thresholding). It may be that with the limited data set size, the use of Oversampling has also helped improve accuracy by increasing the number of samples seen by the network in training.

5.5.2.2 Dropout Analysis

When exploring the effectiveness of data augmentation and class imbalance, the dropout rates have remained the same, to observe the effects of these methods in isolation. Whilst dropout rates observed in neural networks are typically in the form of a lower rate for max pooling and higher rate for fully connected, they are still seen to vary, often depending on the length of training time used.

Epochs 10	No Dropout	Dropout 0.1, 0.25	Dropout 0.25, 0.5
Rail Maintenance	**36.66%**	30.00%	26.66%
CIFAR-10	**34.54%**	33.47%	29.32%
Epochs : 25			
Rail Maintenance	40%	**46.66%**	40%
CIFAR-10	60.12%	61.23%	**61.43%**
Epochs: 50			
Rail Maintenance	46.66%	**63.33%**	**63.33%**
CIFAR-10	46%	**64.54%**	63.89%

Figure 18: Network B testing accuracy for various dropout rates and epochs

Observing the accuracy of Network B when trained for a different number of epochs with varying dropout rates, the rates do not have a consistent effect on overall accuracy. In general, it has been observed that over a higher number of epochs, the employment of a dropout rate has yielded a greater overall accuracy, due to dropout regularisation's effectiveness in reducing over-fitting over extended training time. However, with very short training time (10 epochs) dropout rates have led to a decrease in overall accuracy. It may be that with shorter training, removing neurons from the network at each epoch via dropout is not allowing each neuron sufficient time to adjust weights accordingly.

5.5.3 Comparison of Network A and B

Design A				Design B			
Missing	**No Issue**	**Obscured**	**Predicted/Actual**	**Missing**	**No Issue**	**Obscured**	**Predicted/actual**
3	1	6	**Missing**	6	3	1	**Missing**
4	3	3	**No Issue**	2	5	3	**No Issue**
2	0	8	**Obscured**	1	1	8	**Obscured**

Figure 19: Confusion Matrices for highest performing Network A (Left) and Network B (Right)

Studying the confusion matrices in Figure 19 allows for further analysis on the classification powers of both Network designs. It can be seen that Design B has shown greater recall of both missing and no issue class images, with both networks having equal recall on the obscured class. With that being said, the increased recall of no issue fasteners observed in Design B has come at the cost of the precision score for this class, with a larger number of test samples being misclassified as no issue.

Overall, it can be asserted that design B has clearly produced a higher performing model, achieving better results across almost all performance metrics. Design A's overall accuracy is due in a large part to its recall of 80% obscured fasteners, which makes up over half of correctly classified samples, whilst having a low precision score for the class (less than half of samples classified as obscured in Design A were correct). It seems that Network A's lack of parameters has prevented the network from being able to capture distinguishing features of class samples effectively, whilst Network B has shown limited abilities in classification.

Network A	Precision	Recall	F1- Score
Missing	0.33	0.3	0.316
No Issue	0.25	0.3	0.275
Obscured	0.47	0.8	0.635
Overall/Average	0.35	0.46	0.41
Network B	Precision	Recall	F1- Score
Missing	0.67	0.6	0.63
No Issue	0.56	0.5	0.53
Obscured	0.67	0.8	0.73
Overall/Average	0.63	0.63	0.63

Figure 20: Performance Metrics for best performing Network A and Network B

5.6 Transfer Learning

Whilst further increases to network depth are likely to see more accurate models being produced, the lacking data set size and training time available leads to the conclusion that continuing to train Naïve CNNs is unlikely to produce the best possible results for the project. As such the focus of the network training and evaluation was brought to the area of transfer learning. Various pretrained CNN architectures have been reconfigured for the project's classification task, with different degrees of fine tuning evaluated. Each network has been modified by having the original SoftMax output layer removed. In place of the SoftMax layer, two full connected layers have been implemented for the network to learn features specific to the maintenance issues, and a new SoftMax output layer added containing 3 neurons, representing the network's output probabilities for each class. The Learning rate has been reduced to 0.0001, as it is expected that the pretrained weights already hold a good understanding of features, and large changes in these weights during training may result in a loss of useful feature detection.

5.6.1 VGG-16

VGG-16 was the most accurate model implemented in the ILSVRC 2014 challenge by Oxford Universities Visual Geometry Group (Simonyan & Zisserman, 2015). The network architecture and the optimal weights used in the competition have been loaded into the development platform. From here the network has been re-trained for the new task with different levels of fine tuning (adjusting the layers which may update weights through training) to study how the network's accuracy is affected.

Layers 1-8 Frozen (VGG-16 A)	Layers 1-7 frozen (VGG-16 B)	Layers 1-5 Frozen(VGG-16 C)	Layers 1-3 Frozen (VGG-16 D)	No Layers Frozen (VGG-16 E)
63.33%	63.33%	**76.66%**	**76.66%**	**76.66%**

Figure 21: Accuracy Scores for VGG-16 Transfer learning, various levels of fine tuning Epochs:10

The method of transfer learning led to new maximum accuracy, with networks employing moderate fine tuning matching the best accuracy score seen in Network B, with significantly fewer epochs (although training time was significantly increased). The highest levels of accuracy were seen in the networks employing deeper fine tuning, with 3 out of 5 models achieving the new high score of 76.66% accuracy.

Observing the historical training and validation accuracy of the network it can also be seen that despite the removal of dropout regularisation, the network has not exhibited any symptoms of overfitting, with both training and validation accuracy being comparable throughout training.

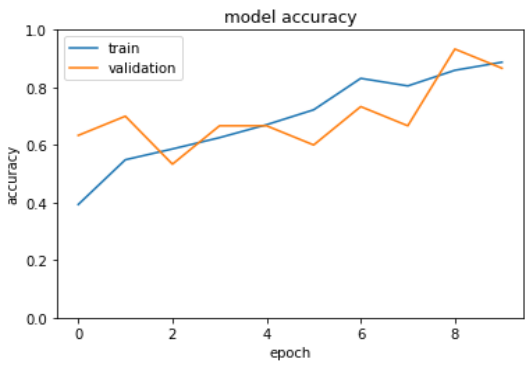

Figure 22: Historical accuracy for VGG-16 throughout training

VGG-16			
missing	no_issue	obscured	predicted/actual
5	5	0	missing
0	10	0	no issue
2	0	8	obscured

Figure 23: Confusion Matrix for VGG-16, all 3 models achieving highest accuracy produced the same output predictions.

What was particularly interesting was that by employing transfer learning to VGG-16, the No Issue class saw the most significant increases in recall and precision, with 100% recall of this class observed. This class seemed most problematic for the previously trained naïve models, yet for the retrained VGG-16 network this appears to be the networks strongest classification ability. Similar performance scores were observed in the missing and obscured class when compared to Network Design B.

5.6.2 Inception_V3

The inception V3 Architecture is a slightly modified version of the Inception V2 architecture that produced the competition winning results in the ILSVRC 2014 (Szegedy, et al., 2015). The modifications made to the network were outlined in their follow up paper (Szegedy, et al., 2016).

Due to the huge depth of the InceptionV3 network (over 100 layers in the network), it was not viable to perform training at all possible levels of fine tuning. Based on the previous results found with the transfer learning of the VGG-16 network, coupled with the findings in other papers (Tajbakhsh, et al., 2017), multiple instances of the network have been trained with deeper levels of fine tuning.

Accuracy of InceptionV3		
Layers 1-5 Frozen	Layers 1-3 Frozen	No Layers Frozen
50%	56.66%	**83.33%**

Figure 24: Accuracy scores of InceptionV3 Architecture employing various levels of fine tuning

The transfer learning performed on the InceptionV3 network has led to a new record accuracy for the project, with 83.33% accuracy achieved at testing when all layers of the network have been finely tuned. Interestingly the network's accuracy was significantly lower even when a very moderate number of layers had their weights frozen during training. It may be that with the complex nature of the inception modules employed in the network, freezing earlier weights has unforeseen effects on the effectiveness of the network's training.

Inception V3			
Missing	No_issue	Obscured	Predicted/Actual
8	2	0	Missing
0	10	0	No_issue
3	0	7	Obscured

Figure 25: Confusion Matrix output for InceptionV3

The network architecture displayed a much higher level of recall in the Missing class compared to the VGG-16 architecture, as well as a higher level of precision in both the No Issue and missing classes.

5.6.3 ResNet-50

ResNet-50 was one of the Networks developed by the Microsoft Research Asia team, employed as part of the ensemble of networks that won the ILSVRC 2015 challenge (He, et al., 2016). Whilst the ResNet-152 network produced the highest standalone accuracy in the paper, training such a large network was unfeasible, and accuracy of both these network designs were comparable (ResNet50: 94.75% Top 5 Accuracy, ResNet-152: 95.51% Top 5 Accuracy). The same focus was made on deeper levels of fine tuning as seen in the InceptionV3 Architecture.

ResNet-50		
Layers 1-5 Frozen	Layers 1-3 Frozen	No Layers Frozen
73.33%	86.66%	**86.66%**

Figure 26: Accuracy Scores for ResNet-50. Epochs:10

ResNet-50 has delivered new accuracy high scores at testing compared to previous models, delivering the highest recall of missing class samples, and precision of no issue class samples. Despite the network achieving the greatest overall accuracy, it has a lower recall of obscured fasteners compared to the VGG-16 model, which managed to best the recall rate whilst keeping perfect precision scores.

ResNet-50			
Missing	No_issue	Obscured	Predicted/Actual
9	1	0	Missing
0	10	0	No_issue
3	0	7	Obscured

Figure 27: Confusion Matrix output for ResNet-50

5.7 Ensemble

In line with the further increases to accuracy seen in ensembling neural networks (He, et al., 2016), and in reaction observed strengths of the three network architectures explored, various ensembles of the neural networks were implemented, where the ensembled output was created both by taking the arithmetic mean and the harmonic mean of the networks' combined output layers to produce the ensembled networks' classification, to see if further accuracy improvements could be made.

	VGG16+Incep	VGG16+ ResNet	ResNet+Inception	VGG+ResNet+Inception
Arithmetic Mean	86%	**90%**	86%	86%
Harmonic Mean	83%	83%	83%	86%

Figure 28: Accuracy of ensembles

.

In the case of ensembling the networks via the arithmetic mean of SoftMax outputs, further increases in accuracy have been seen in the combination of VGG-16 and ResNet-50, with the remaining 3 combinations meeting previous accuracy scores, each showing high accuracy, and matching the performance of the best performing ResNet-50 network seen earlier in the report. The use of Harmonic mean however has not led to any further improvements in accuracy, potentially as vastly different values in networks SoftMax outputs would lead to an overall smaller value being output by the ensemble, mitigating the ensembles ability to make use of both networks relative strengths.

Interestingly, it is the combination of VGG16 and ResNet-50 that have together produced the best scores, despite VGG-16 being the least accurate network of the three when implemented as a singular network. This combination has allowed VGG-16's strength in recall of obscured class samples to enhance the overall classifications of the ResNet-50, without causing a detrimental effect to other classification capabilities.

6 Implementation into GUI program

The final application takes the form of a GUI interface, built using Pythons TKinter package. It can be used to show the proficiency of the trained CNNs in detecting the maintenance issues explored in the paper.

The program implements the best performing ensemble created through the design and analysis of the neural networks and allows for an image to be selected from a local directory and then classified.

The user selects the image it wishes to classify by means of the browse image button. Once selected the image and its directory are both displayed in the GUI. Once the images have been selected, the user presses the classify button, passing the input images to each of the neural networks which in turn produce a SoftMax output.

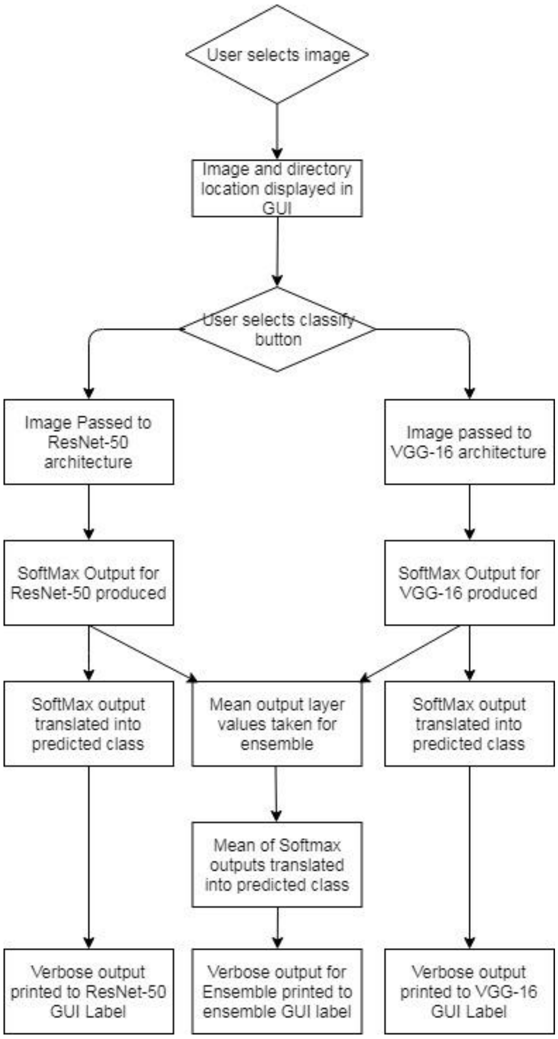

Figure 29: Flow chart illustrating applications functionality

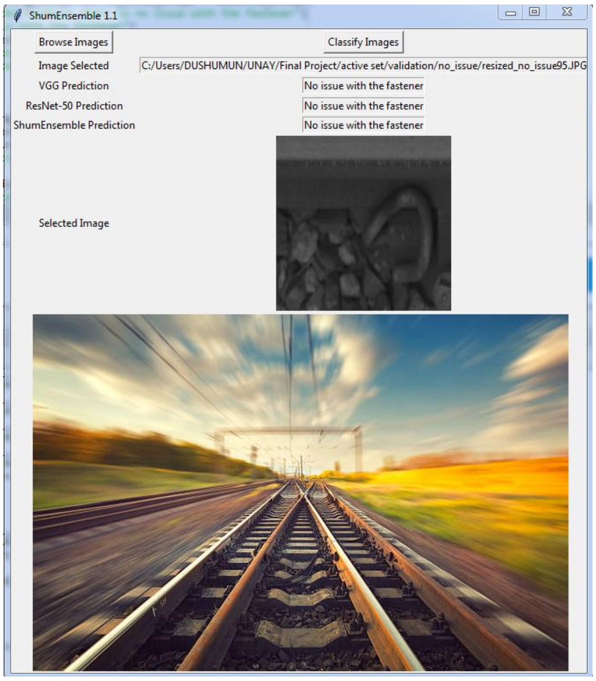

Figure 30: Screen shot of GUI making prediction on selected image

The mean of both networks' output layers is calculated, creating the Ensembled output classification. The program then translates both the individual SoftMax layers and the ensembled prediction, returning a verbose output for all three predictions to the GUI, hiding the complexity of the system and allowing a person with no knowledge of CNNs to understand the prediction.

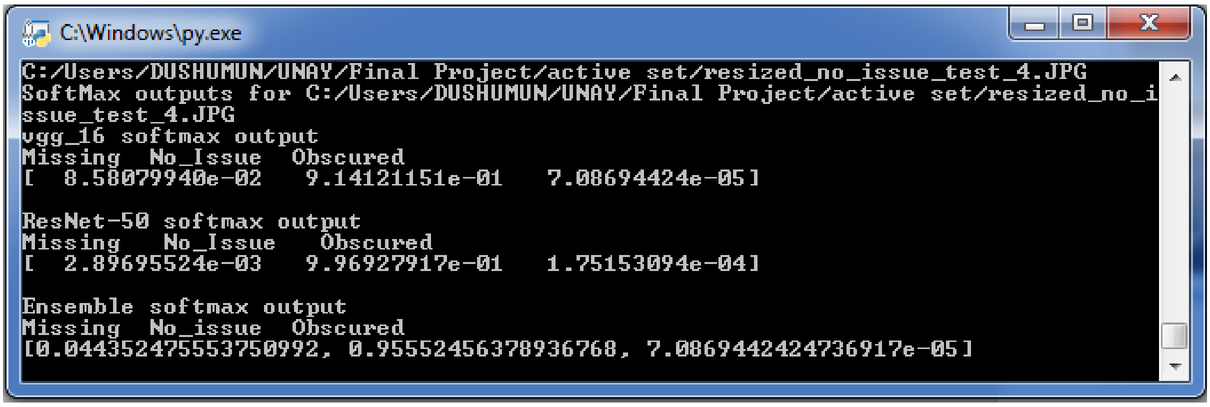

Figure 31: Command window output displaying networks SoftMax outputs

The SoftMax values from each neural network are also printed to the command prompt, allowing for a more in-depth analysis of the system's performance.

Installation of the program can be completed simply by extracting the zipped file containing the necessary files to the desired location. Once extracted, selecting the file SHUMENSEMBLENET 1.34.py will start the program with full functionality.

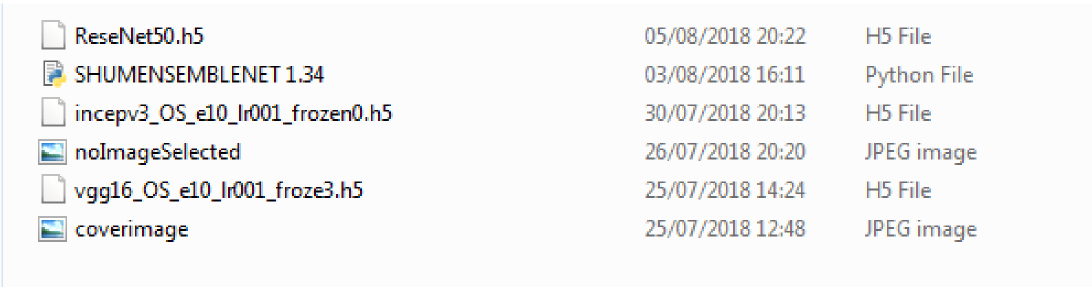

Figure 32: Files required in directory for running of the GUI program

To run the application via a Windows PC, the following prerequisites are required.

Python 3.5 installation with the following libraries.

- TensorFlow
- Keras
- Numpy
- TKinter

7 Testing

In order to ensure that the networks deployed into the program retained their training weights and classification abilities, the SoftMax outputs from initial testing, as well as the SoftMax output produced once the networks were implemented into the GUI program where compared. Each output matched exactly in both cases, proving that the networks weights and architecture had stayed consistent when imported into the GUI, and that images are being processed correctly within the application.

To test the effectiveness of the ensemble, images were input into the GUI that had been misclassified by one of the networks, whilst correctly classified by another, to see if the ensemble would resolve this issue.

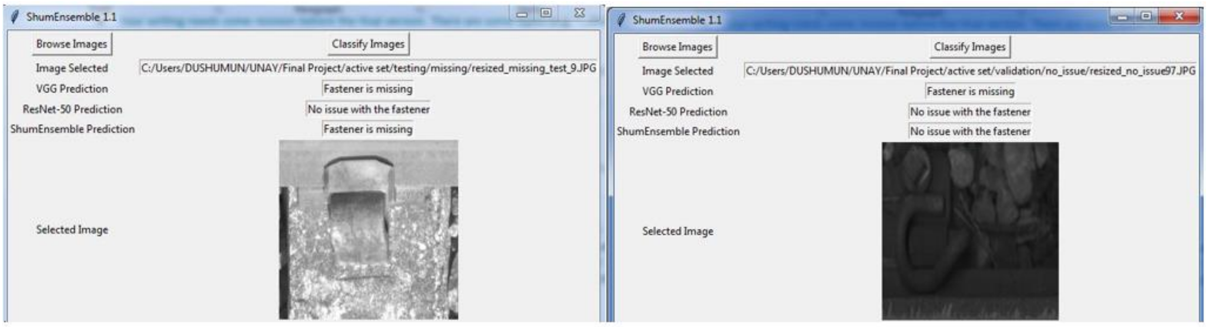

Figure 33: Examples of how the ensemble has overcome the incorrect classification individual Networks

Observing the screen capture, in the examples on the left it can be observed that the correct VGG-16 prediction has overcome the ResNet-50 misclassification in the ensemble (the fastener is missing). This is also seen in the right-hand screenshot where the ResNet-50 Network has helped correct the VGG-16 error as part of the ensemble.

8 Project Management

8.1 Project Schedule

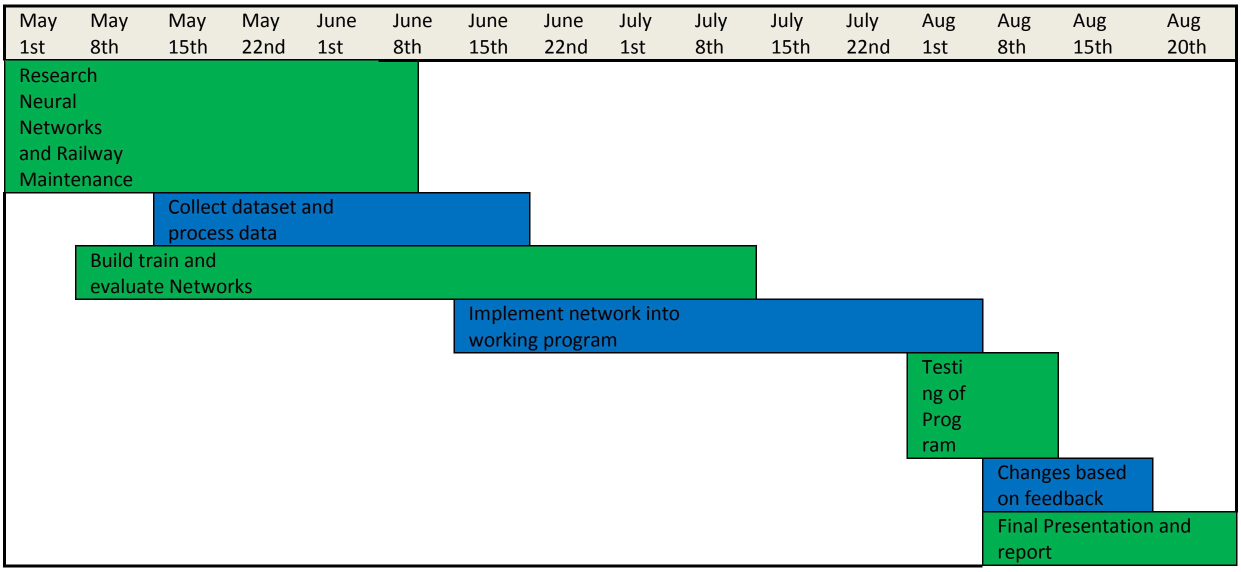

Figure 34: Gant Chart produced at the beginning of project.

The Gant chart shows the initial project schedule laid out at the start of the project, that was made to break the project down into more manageable sections, allowing for effective time management, and to keep focus over the long period that the project ran for. Whilst the Gant chart was valuable in the management of time, it was not followed to the word for several reasons.

Whilst a significant time was spent researching the related fields in the beginning of the project, challenges faced later in the project, particularly those relating to the structure of the formed data set, led to further research into the area of CNNs to tackle problems, particularly those of class imbalance seen in the data set. Such challenges were unforeseen in the initiation of the project due to the data set not being procured and backtracking to earlier stages was necessary in order to produce the best results during the design phase.

The creation of the report also became a more prolonged affair, due to the research nature of the project, creating the full report at the end of the projects time frame may have led to the misplacement of valuable information gathered throughout the research. As such sections were worked on throughout the project, with the final 4 weeks of the project dedicated to bringing the sections together into a suitable format.

8.2 Risk Management

The most standout risk to a project such as this is data loss. With word documents, spreadsheets, programs as well as saved neural network architectures, storing the project files in a single place was not feasible.

All none sensitive files were backed up both physically on an external hard drive and on Google Drive’s cloud storage platform. The data set provided was backed up on the same external hard drive as well as the One Drive cloud storage provided by the University. Although the data set was provided with no confidentiality agreement due to it containing no trade secrets or personal information, the external hard drive directory containing the set was encrypted using Windows built in encrypted file system to ensure in the event of theft when travelling with the hard drive that it would remain out of unintended hands.

With the off-site trip made to meet with Omnivision, the dates and times were communicated to both the project supervisors and next of kin. It was arranged to make contact with next of kin via phone before and after the trip. This ensured the relevant people would be aware of a potential emergency.

The project suffered a hard drive failure during its work, however due to the measures taken in backing up the projects files, the loss of work was minimal and caused marginal disruption to the overall project.

9 Critical Appraisal

Throughout the training of the different neural networks in the project, a correlation between increased training time and accuracy has been observed, with the highest performing network, ResNet-50 taking the longest of all networks to train with nearly 10 hours training time. The prominent outlier to this trend is observed in the InceptionV3 architecture, which achieved the 2nd highest accuracy of all networks trained, whilst taking significantly less time to train than both other networks used in transfer leaning. It was actually similar in training time to the original naïve models, whilst exhibiting far greater accuracy scores.

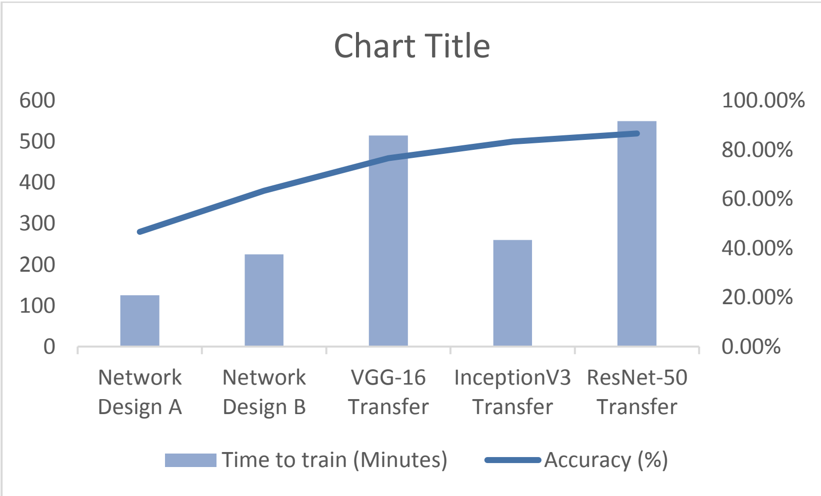

Figure 35: Best accuracy achieved by different network architectures and their training times

Whilst the training time of each network has still been reasonably manageable, with even the longest training times not exceeding 10 hours, a more extensive data set would require further increases to training. Even with the implementations of GPUs to increase the efficiency of training, the InceptionV3 architecture may be a more desirable choice for such scenarios, due to its more efficient use of parameters.

The training of the networks has demonstrated transfer learning's effectiveness in producing accurate image detection models when compared to training naïve models. Making use of previously trained architectures and their weights allows for the training of highly accurate models where only small data sets are available, saving time in training and evaluating numerous different custom architectures. It has also been observed in this case that the deep fine-tuning of all layers in the networks implemented for transfer learning led to higher levels of accuracy compared to those networks that fine-tuned more moderate layers. It seems that due to the large differences in image classification problems observed between the ImageNet data set and the Rail Maintenance data set, that allowing for the adjustment of even the high-level features captured in earlier layers allows for the best results in re-purposing these networks.

Whilst applying transfer learning to these sophisticated neural networks has yielded high classification accuracy, the computational cost of storing and implementing the networks is significantly higher.

Network	Design A	Design B	VGG-16	InceptionV3	ResNet-50
File Size	25mb	40mb	263mb	431mb	204mb

Figure 36: File sizes for storing network architectures and weights

The file sizes for each network displayed in Figure 36 illustrate the difference in storage required for each network. As networks need to be loaded into memory in full in order to be implemented, this also represents a large difference in the computational power required for each network. The storage requirements of VGG-16, InceptionV3 and ResNet-50 architectures are much larger than the naïvely trained models, leading to greater memory requirements for the hardware intended for implementation. Whilst none of these represent prohibitive size in terms of implementation on a desktop computer, smaller computer systems that could potentially be implemented onto drones for classification whilst out in the field may not be possible, especially if an ensemble of these networks was intended for implementation. It may be that through further training of smaller naïve models on more extensive data sets, higher levels of accuracy can be achieved, whilst gaining a better ratio of accuracy against hardware requirements.

The best performing networks trained in the project have led to highly accurate models, with the most accurate ensemble of networks achieving 90% classification accuracy in testing. Such results show the promise of implementing CNNs into future computer vision systems with the purpose of rail maintenance detection.

VGG+ResNet-50	Precision	Recall	F1-Score
Missing	0.75	0.9	0.818181818
No Issue	0.909090909	1	0.952380952
Obscured	1	0.8	0.888888889

Figure 37: Performance Metrics for VGG-16 and ResNet-50 ensemble

When observing the performance metrics of the best ensemble, there has been a great improvement on the precision of each classification when compared to the current Plain Line Pattern Recognition system currently employed on the New Measurement Trains (which currently achieves only 33% classification precision). However, with less than perfect recall of the maintenance issues, further measures would need to be put into place to tackle the occurrence of false negatives. The number of maintenance issues explored for the fasteners is also far from exhaustive, with classifications such as broken housing and retracted fasteners being excluded from the training due to an insufficient number of samples. In fact, the data set used in the project has been extremely lacking in size overall. Although good accuracy has been achieved, further improvements could almost certainly be seen with a larger data set, and this would also lead to more feasibility in training for the excluded classes. A larger testing set could also be produced, which would allow for more confident conclusions derived from the accuracy seen at testing. With just 10 testing samples each, the results of testing are far from conclusive, although still encouraging.

A range of architectures and regularisation methods were explored in the project which led to good results. However, there is still many modifications that have been left unexplored. Different optimizers may have led to findings in increased training efficiency and even higher accuracy of results, however the results found in the project were still of high accuracy, with manageable training times. A more thorough analysis on different image augmentation techniques could have also been performed, however with such a wide range of techniques available it could have easily taken up a disproportionate amount of time, reducing the analysis on other techniques that also led to significant accuracy improvements.

Whilst the GUI application allows for a good demonstration of the CNNs' capabilities in the classification task, it lacks a true industry purpose. With the implementation of a single image at a time, manual inspection would likely be faster than inspection via the application. As such this application would need to be thoroughly extended for use in industry.

10 Conclusions

10.1 Achievements

The work in the project has demonstrated the great potential in the implementation of CNNs to enhance the computer vision technology used to detect maintenance issue in railways. Through literary research and meeting face to face with industry professionals, current systems and processes and been identified, and the potential for improving such systems has been identified. This research has led to CNNs being trained on a data set containing different maintenance issues present in Railways, achieving high levels of accuracy through the evaluation of different architectures and regularisation methods. The design and analysis by which the best accuracy was achieved has offered great insights into design choices that may help guide the industry in its own deep learning research. The final application demonstrates not only the classification powers of CNNs, but also how the technology can be implemented into an application that can be used by those with no expertise in machine learning techniques.

10.2 Future Work

Whilst the work done in this project has successfully demonstrated the potential for implementing CNNs to tackle the problem of detecting and classifying maintenance issues, the networks' capabilities to identify a variety of issues is very limited. With an extended data set containing larger samples of different maintenance issues it may well be possible for the network to be extended to classify a far greater number of maintenance issues, with further improvements to overall accuracy.

Whilst the system in place can produce an accurate output classification from an input image, it still requires manually selecting individual images for the classification to occur. The system could be extended to except multiple images, and automatically produce a report that highlights the potential issues found across the input images. This would extend the application from a technology demonstration to a potential tool for direct use in industry.

The images input in the model have also been acquired by manual extraction from the original source images, which initially consisted of sections of rail line with multiple fasteners. By utilising separate CNNs for the use of segmenting the sleepers and rail in the image, it may be possible to automate the process of gathering the individual fastener images from the raw rail images gathered by the NMT fleet (fasteners are present on each sleeper on both sides of the rail when in good working order). These could then be passed to classification CNNs to deduce the condition of the fastener, producing a fully automated system from raw image collection to classification. The apparent challenge faced by this is that sleepers can often become completely obscured by ballast, therefore making the task of reliably detecting every sleeper in the rail network difficult.

Figure 38: Sample image from data set containing completely obscured sleepers and fasteners (leftmost sleepers)

Whilst a CNN may not be able to reliably detect all sleepers in sections of railway when implemented alone, it may be possible to implement a CNN into a hybrid system, which makes use of the standardised distance between sleepers on a track to overcome this challenge.

British Rail lines have a standardised spacing of 700mm on straight lines and curved tracks with a radius over 600 metres, and a smaller 650mm spacing between sleepers on tighter curves with a radius below 600 metres (Defense Movements & Transport Policy Division, 2009). Taking this information and applying it to a system that could work alongside a CNN trained for image segmentation, a system could be built that would be able to predict the expected location of the subsequent sleepers based on the location of those that have been previously detected. By doing so the system would be able to fill in the expected sleeper locations where obscured and collect the necessary images required. Taking advantage of the gyroscopes and accelerometers available on board the fleet of measurement vehicles, the system may also be able to detect the degree of the curve in the track, allowing the system to adjust the distance for the next expected sleeper.

Looking beyond enhancing the capabilities of current technologies enabled by the NMTs, exploring more complex sections of the rail network such as crossings and junctions could bring greatly extended automation to the rail maintenance process. Whilst the NMT fleet currently is unable to perform maintenance detection on these sections due to their complex nature, implementing new technologies such as autonomous drones in order to collect images at a larger number of angles, as well as CNNs to perform classification on such images, this area of maintenance inspection currently performed manually may be able to see significant improvements to efficiency. Whilst it has been previously mentioned that the hardware requirements for complex CNN implementations may make their implementation directly onto a drone unfeasible, switches and crossings make up relatively small sections of the rail network. As such the installation of localised processing units at each of these sections of the rail network maybe possible, that could allow the implementation of the more complex networks, and therefore maximising the classification powers. This would allow the drones to collect the relevant images and then transmit these images to the localised processing unit to detect the presence of maintenance issues.

Figure 39: Could autonomous drones lead to the future of rail inspection?

11 Personal Reflections

Through the research undertaken in this project and the training and implementation of the neural networks on the chosen task, I have developed a huge wealth of knowledge both in the advance architecture designs of cutting edge convolutional neural networks and gained skills in implementing such designs in a real-world scenario. The time constraints in the project, coupled with the lacking computational power somewhat restricted the amount of research that could have been undertaken, particularly regarding the absence of powerful GPU's that would have led to vastly increase training times of the CNNs explored. However, high levels of accuracy in the classification were achieved, and the learning of methods to increase this accuracy have given invaluable knowledge in tackling a range of data set problems.

Whilst these skills were partially developed by the implementation of CNNs to tackle well known classification problems such as the CIFAR-10 data set, used during initial analysis of methods and parameters in the project, I have found that by gathering the raw data set acquired from the industry, processing it into a format that allows for training, and tackling the sets various limitations and challenges, invaluable skills have been gained which will directly translate into potential careers with practical industry purposes.

More abstractly, I have developed skills in observing a research area with a wide breadth of potential areas to explore, and through research and discussion with my project supervisors, as well as contacts I have made in the industry, throughout the project I have been able to bring my focus to a specific task in order to produce a small but significant contribution to the field of research.

The skills gains have also extended past just the training and evaluation of the networks. By creating a GUI program that showcases the implementation of multiple neural networks, I have gained knowledge in how such technologies could be implemented into practical applications. The program delivered can also act as a demonstration of my skills and abilities, showing my abilities to self-learn and work towards a specified goal with limited help and guidance. The skills I learnt and applied to combine the data science research into workable program is something I am particularly proud of, being a student with a fundamentally mathematical background and a limited knowledge of software development.

12 Bibliography

Ballard, D. & Brown, M., 1982. *Computer Vision.* 1st ed. s.l.:Prentice Hall.

Buda, M., Maki, A. & Mazurowski, M., 2017. *A systematic study of the class imbalance problem in convolutional neural networks,* [online]: ArXiv.

Defense Movements & Transport Policy Division, 2009. *MOD UK railways permanent way design and maintencance: policy and standards.* [Online]
Available at:
https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/565742/MOD_UK_Railway_Permanent_Way_Design_and_Maintenance_-_issue_4.pdf
[Accessed 23 07 2018].

Denham, C., 2012. *The Future of Continuously Welded Track Inspection is here.* [Online]
Available at: https://www.networkrailmediacentre.co.uk/news/the-future-of-continuously-welded-track-inspection-is-here
[Accessed 26 06 2018].

Department For Transport, 2017. *Rail Factsheet: November 2017,* London: Department For Transport.

Deshpande, A., 2016. *The 9 Deep Learning Papers You Need To Know About.* [Online]
Available at: https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html
[Accessed 17 05 2018].

Gershgorn, D., 2017. *The Data that Changed the Direction of AI Research and Possibly the World.* [Online]
Available at: https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/
[Accessed 4 June 2018].

He, K., Zhang, X., Ren, S. & Sun, J., 2016. *Deep Residual Learning for Image Recognition.* Las Vegas, IEEE.

Huang, T., 1996. *Computer Vision: Evolution and Promise.* Egmond ann Zee, The Netherlands, CERN. Geneva, pp. 21-25.

Krizhevsky, A., Sutskever, I. & Hinton, G., 2012. *ImageNet Classification with Deep Convolutional Neural Networks.* Seattle, ACM.

Lecun, Y., Bottou, L., Yoshua, B. & Haggner, H., 1998. *Gradient Based Learning Applied to Document Recognition.* San Francisco , IEEE.

Network Rail, 2017. *New Measurement Train.* [Online]
Available at: https://www.networkrail.co.uk/running-the-railway/looking-after-the-

railway/fleet-machines-vehicles/new-measurement-train-nmt/
[Accessed 2 07 2018].

Ng, A., 2018. *Stochastic Gradient Sescent.* [Online]
Available at: https://www.coursera.org/lecture/machine-learning/stochastic-gradient-descent-DoRHJ
[Accessed 13 07 2018].

Ng, A., 2018. *Why do residual networks work?.* [Online]
Available at: https://www.coursera.org/lecture/convolutional-neural-networks/why-resnets-work-XAKNO
[Accessed 15 07 2018].

Nomad Digital, 2017. *Rail industry survey: Major Challenges facing rail operators, maintainers & owners and the role of ICT,* [online]: Nomad Digital.

Papert, S., 1966. *The Summer Vision Project,* Massachusetts: Massachusetts Institure of Technology.

Perez, L. & Wang, J., 2017. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning,* [online]: ArXiv.

Rail Magazine, 2003. Doctor Yellow. *Rail Magazine,* July, pp. 40-45.

Simonyan, K. & Zisserman, A., 2015. *Very Deep Convolutional Neural Networks for Large-Scale Image Recognition.* Boston, IEEE Conference on Computer Vision and Pattern Recognition.

Srivastava, N. et al., 2014. Dropout: A Simple Way To Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research,* June, Issue 15, pp. 1929-1958.

Szegedy, C., Liu, W., Jia, Y. & Sermanet, P., 2015. *Going Deeper with Convolutions.* Boston, IEEE Conference on Computer Vision and Pattern Recognition.

Szegedy, C. et al., 2016. *Rethinking the Inception Architecture for Computer Vision.* Las Vegas, IEEE.

Tait, S., 2018. *Engineering Manager* [Interview] (7 June 2018).

Tajbakhsh, N., Shin, J. & Gurudu, S., 2017. Convolutional Neural Networks for Medical Analysis: Full Training or Fine Tuning?. *IEE Transactions on Medical Imaging,* Issue 35, pp. 1299-1312.

The Railway Magazine, 2003. NR's yellow HST enters service. *The Railway Magazine,* July, p. 9.

Viola, P. & Jones, M., 2001. *Rabid object detection using a boosted cascade of simple features.*. Kauai, ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001.

Wang, Y., 2014. An Analysis of the Viola Jones Face Detection Algorithm. *Image Processing Online,* Issue 4, pp. 128-148.