Robust Image Classification using Deep Learning

Janith Thanthulage; B523685

Electrical and Electronic Engineering

**Abstract:** I WILL DO THIS AFTER WRITING THE SUMMARY… summarising the summary

The abstract is a short summary that should state the objectives of the project described in the paper and include brief conclusions such as any newly observed experimental results. The abstract should contain no more than 200 words and should be self-contained, i.e. no references should be included and readers should not have to read the paper to understand it.

# Introduction

Recent years have seen the increased use of computer vision in a multitude of different disciplines and contexts. More specifically, systems have been implemented with the intention to leverage the autonomy and accuracy that can be achieved by image recognition systems for complex tasks related to, but not limited to, medical diagnosis, security, autonomous driving and . [1-5]

A paradigm shift in the car manufacturing world can be witnessed as a growing number of manufacturers utilise image recognition and classification systems to allow for features such as automated parking, driving and alerting drivers of possible collisions on the road. [3] Additionally, computer vision systems have become an agent in the medical field to aid with biomedical research such as its use in recognising the prevalence of heart disease in samples instantaneously. [4]

As we become more dependent on such systems, the ability to robustly withstand possible attacks to the integrity of such systems goes from being an afterthought to a necessary point of focus. Szegedy et al. first generated small perturbations on images used in an image classification task and was able to thoroughly demonstrate vulnerabilities in classification systems by fooling such neural networks with high probability. [7] These samples capable of causing such a misclassification were named as Adversarial Examples.

Consequently, this point of attention has in more recent years gathered an increasing amount of interest in both research and industry. The multifaceted task of testing whether computer vision models are robust remains a great challenge, arguably due to the fact that measuring robustness cannot be given by a binary answer. Hence as a result, many network architectures currently in use may show vulnerabilities due to both improper training combined with a lacking of proper testing to measure the specific robustness to targeted attacks.

# Adversarial Attacks

An adversarial attack is a deliberate attempt at deceiving a machine learning algorithm such as a Convolutional Neural Network (CNN) or Deep Neural Network (DNN). This is implemented through the use of malicious adversarial examples that are carefully crafted by an attacker to force the misclassification of a model to a particularly chosen input or through the addition of purposeful noise perturbations added to an input to mask relevant features and highlight others (again causing misclassifications). Note that adversarial samples are created at test time, after the DNN has already been trained by the defender, so do not perform any modification to the training process.

Impressive research has been conducted in performing attacks to cause misclassifications in the contexts detailed. One such work was the creation of perturbations to mimic graffiti that may exist on traffic signs and “thus hide in the human psyche.” [6] By creating such perturbations, it was found possible to misclassify objects such as a Stop sign to a Speed Limit 45 sign. It can be seen how, in the context of autonomous vehicular navigation, this is particularly dangerous.

Most research conducted has focused largely on the space of image classification systems due to perhaps the easiness of visualising the perturbations. The research conducted in [8] demonstrates the vulnerability of neural networks in a much more general application. The work conducted, demonstrated the ability to “construct targeted audio adversarial examples on automatic speech recognitions” systems: more specifically it demonstrated the ability to cause misclassifications on Mozilla’s DeepSpeech model. This emphasised the existing vulnerability of neural networks in a more general application which may otherwise have been overlooked further prompting the need for a methodology of defence against adversarial examples.

It can be seen how the threat of adversarial attacks have become an ever-increasing threat to autonomous systems dependant on trained machine learning networks. Despite being trained in secure environments and the networks themselves being contained privately; models still show susceptibility to adversarial perturbations.

# aim

This work aims to build upon and compare the effectiveness of existing methods of making a classification network robust.

Existing methods of defence against adversarial attacks can be generally split into two broad categories. One methodology is based around the robust optimization of a model, where a defender will explicitly incorporate knowledge of potential attacks into a model’s training stage in addition to its usual training and testing. Adversarial training is a common utilisation of this methodology as to make an existing model more robust [REFERENCE]. It has been shown by Goodfellow et al. that the retraining of a compromised network using its adversarial examples can have the benefit of providing an additional regularization beyond that provided by using methods such as dropout [9].

The other methodology is based around the idea of developing an architecture that is able to withstand or perform a level of rejection against adversarial examples through the identification of possible outlying perturbations. Such methods often utilise a form of model dimension reduction or model gradient smoothing [REFERENCE]. A recent method proposed has been that of Deep Neural Rejection (DNR), an extension of a previously proposed Neural Rejection (NR) method by Melis et al. [10], which makes use of RBF Support Vector Machines (SVM) trained on the higher level features of the compromised network with the objective of withstanding or alternatively rejecting adversarial images [11].

The effectiveness of these methods will be tested specifically against a projected-gradient-defence (PGD) type evasion attack of varying severities, see Section 2.5. The severity or strength of the attack (ε) will demonstrate the Euclidean distance (or L2 norm) of the adversarial image from the original: the greater the distance, the greater the perturbation and therefore the larger the expected possibility for misclassification. Hence, a measure of the robustness of a network or defence of a network can then be attributed to the attack strength required to cause its misclassification.

# methodologies

# dataset used for implementation

To demonstrate the effectiveness of attacks and defences, the MNIST database was used. It is a combination of two of NIST’s (National Institute of Standards and Technology) databases, Special Database 1 and Special Database 3, which consists of handwritten digits written by high school students and employees of the United States Census Bureau respectively. It was predominately chosen as it is a pre-existing and well known dataset which is commonly used for testing classification systems and methodology. Use of this database offers the added ability to easily visualise the results of adversarial attacks and directly compare against the results of other published research which use the same database.

The MNIST dataset consists of 60,000 training and 10,000 testing grayscale images of size 28x28. Figure 1 illustrates nine example images from this dataset. All images were normalised to a range between 0 and 1 before the training and testing stages by dividing the original pixel values of the images by 255.

As detailed in [11], the base DNN model and various defence architectures were similarly trained on a subset of 30,000 training images from the overall MNIST dataset. The adversarial attack, discussed in Section 2.5.1, was performed on a random selection of 1000 images for each ε tested as to create a dataset of this size consisting of perturbed images. This was then used to quote the accuracies and rejection rates of each of the defence methods as can be seen in FIGURE.

A close up of a piece of paper

Description automatically generated

Figure – Typical example images from the MNIST dataset

# software (and hardware) used for implementation

The models and experiments tested were built using Python. Keras and the Scikit-learn libraries were predominantly used to build and train the DNN and consequent SVMs respectively. The frameworks provided by these libraries are very well optimised and offer an attractive ‘ease of use’ element in addition to its large active community. Moreover, the Keras library offers compatibility with TensorFlow-GPU, meaning the relevant algorithms were able to be computed on a GPU, thus improving its speed by up to 10x. This is particularly important due to the size of the datasets and features that are being trained.

# Deep neural network classifier

Before addressing the methods in which the model was attacked, we can first address the base classifier network. First proposed in [12], this architecture was also used as a control in the DNR [11] and NR [10] defence methods. The use of this base classifier architecture allows for a direct comparison between our findings and the results reported for each respective defence method.

The details of each layer of the network can be seen detailed in Table 1.

Table – Base DNN Classifier

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Operational Layer | | | Number of Filters | Size of each filter | Stride value | Padding value | Size of output |
| Input Image (Normalised) | | | - | - | - | - | 1x28x28 |
| Layer 1 | Convolution Layer | Convolution + ReLU | 32 - | 3x3 - | 1x1 - | 1x1 - | 28x28x32 28x28x32 |
| Layer 2 | Convolution Layer | Convolution + ReLU | 32 - | 3x3 - | 1x1 - | 1x1 - | 28x28x32 28x28x32 |
| Layer 3 | Pooling Layer | Max Pooling | 1 | 2x2 | 2x2 | 0 | 14x14x32 |
| Layer 4 | Convolution Layer | Convolution + ReLU | 64 - | 3x3 - | 1x1 - | 1x1 - | 14x14x64 14x14x64 |
| Layer 5 | Convolution Layer | Convolution + ReLU | 64 - | 3x3 - | 1x1 - | 1x1 - | 14x14x64 14x14x64 |
| Layer 6 | Pooling Layer | Max Pooling | 1 | 2x2 | 2x2 | 0 | 7x7x64 |
| Layer 7 | Inner Product Layer | Fully Connected + ReLU | - - | - - | - - | - - | 200 |
| Layer 8 | Inner Product Layer | Fully Connected + ReLU | - - | - - | - - | - - | 200 |
| Layer 9 | Output Layer | Softmax | - | - | - | - | 10 |

Note: each convolutional layer used a normal distribution for the initialisation of their respective filters.

The Keras library was used to train this classifier using the following parameters:

Table – Learning Parameters for Base DNN Classifier

|  |  |
| --- | --- |
| Parameter | Value |
| Learning Rate | 0.1 |
| Momentum | 0.9 |
| Dropout | 0.5 |
| Batch Size | 128 |
| Epochs | 50 |
| Decay | 1e-6 |

The use of kernel or bias regularisation, or rather lack thereof, was not specified in the original reference materials. As a result, the choice to not apply kernel or bias regularisation to the base DNN model was taken. This choice increases the chance of large weights and biases being learnt by the model which increases the chance of a network to overfit to its data. These large weights and biases are typically ill-advised as they can then be exploited by adversarial attacks; however, due to the nature of the work being conducted, it offers a good method of comparison to how much a model can be improved via the defences proposed.

The fully trained model reported the metrics as stated in Table 3. Figure 2 and Figure 3 further illustrates this over each iteration. As a note, the metrics can seem odd at first interpretation due to the testing metrics being better than that of the training metrics. However, a model trained using the Keras library has two modes: training and testing. Any regularisation mechanisms such as the dropout used are not implemented at test time [13] hence can offer some reasoning for the training metrics being marginally worse than that of the testing metrics.

Table – Summary of trained DNN classifier metrics

|  |  |  |
| --- | --- | --- |
|  | Training Set | Testing Set |
| Cross Entropy Loss | 0.0684 | 0.0462 |
| Classification Accuracy | 98.24 | 98.95 |

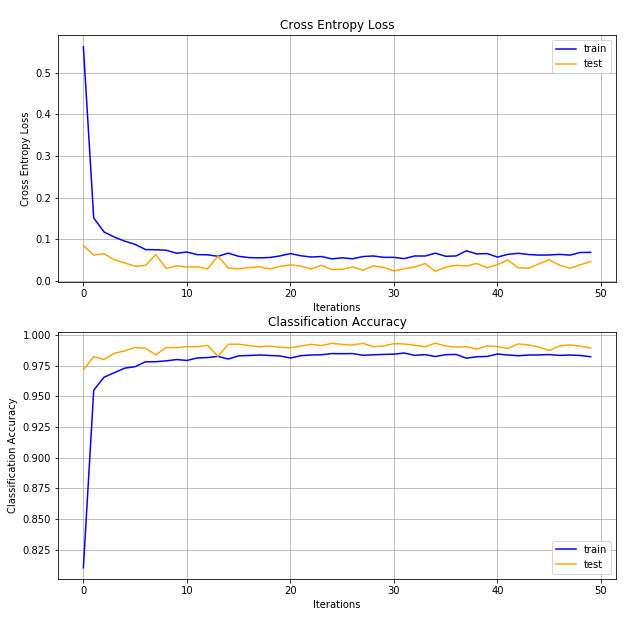


Figure – Cross Entropy Loss

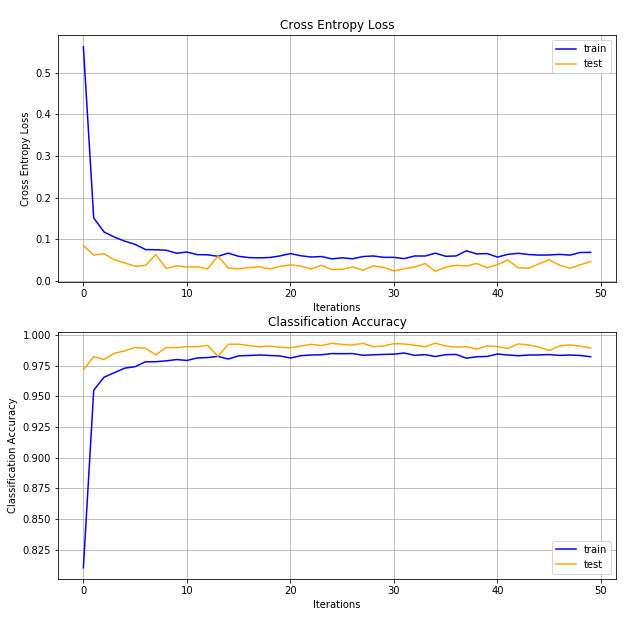


Figure – Classification Accuracy

# defence methods

Each defence architecture labelled below, makes use of multiclass Support Vector Machines trained on one or multiple features of the sublayers within the base DNN classifier and then consequently uses a rejection mechanism to discard possible perturbated samples. The specific layers each defence method is trained can be found in their respective sub-section.

Briefly, given training vectors belonging to two classes and label vector (where n is the number of training data samples and p is the number of features in a data sample), the Support Vector Machine aims to construct a hyper-plane (or set of hyper planes in an infinite dimensional space) to complete the classification task which separates the two given classes accurately. A good separation of the samples from the hyperplane is achieved by maximising its distance to the nearest training data point. This is then equivalent to the minimisation problem:

Introduction of the Lagrange function then allows the conversion of the former minimisation problem into the following dual quadratic optimisation problem as similarly optimised by Scikit-Learn’s SVC function:

C is the regularisation constant which controls the trade-off between a smooth decision boundary and classifying the training points correctly. The regularisation constant, C was chosen through a process of grid search for a satisfactory cross validation score, such that C=0.1. After fitting, the hyperplane separating these two classes can then be defined by the decision function:

Where,

* and are the support vector components found for the optimum fit

For non-linear cases, a kernel function can be introduced to map the features into a higher dimensional feature space as to allow for a better fit of the hyperplane, where the decision function then becomes:

The DNR and NR defence methods both make use of the RBF Kernel defined as such:

Where,

* ‖x-x'‖2 is the squared Euclidean distance between the two feature vectors
* σ is a chosen constant

Scikit-Learn uses the following equivalent kernel which uses the parameter :

All Support Vector Machines trained made use of , where an increasing value of gamma leads to overfitting as the classifier attempts to perfectly fit the data. This was similarly chosen through a method of grid search as C.

A ‘one-vs-rest’ approach was taken in training the SVMs resulting in a separating hyperplane (and thus 10 separate SVMs) being trained for each of the 10 classes. The “decision\_function” method of Scikit-Learn’s SVMs can give a per-class score for each sample by passing the sample (x) through the trained decision function.

The output probabilities of each class (s0…s9) from the Support Vector Machines were then found using Scikit-Learn’s “predict\_proba” function which applies a logistic regression method on the SVM’s decision function score.

# Single layer Neural Rejection (NR)

The architecture proposed as by [12], involves the use of an RBF-SVM trained on the penultimate layer of the base DNN classifier as such illustrated by Figure 4. When training, the features from the relevant layer were extracted and used to train the SVM. The base DNN model remained unchanged during this training process. Note the probabilities of each class as the output, s0…s9, and the rejection mechanism governed by the individual rejection class (sR) set to a constant value, θ: if sR is the highest probability value, the sample is considered to have been rejected.

The implementation of the maximal margin found in SVMs have demonstrated the ability to be very effective in preventing evasion-based attacks; the use of the RBF kernel allows the features to be moved into a non-linear feature-space, thereby making it more difficult for the generation of successful adversarial examples.

The classification accuracy achieved by this SVM can be seen illustrated in Figure 6.

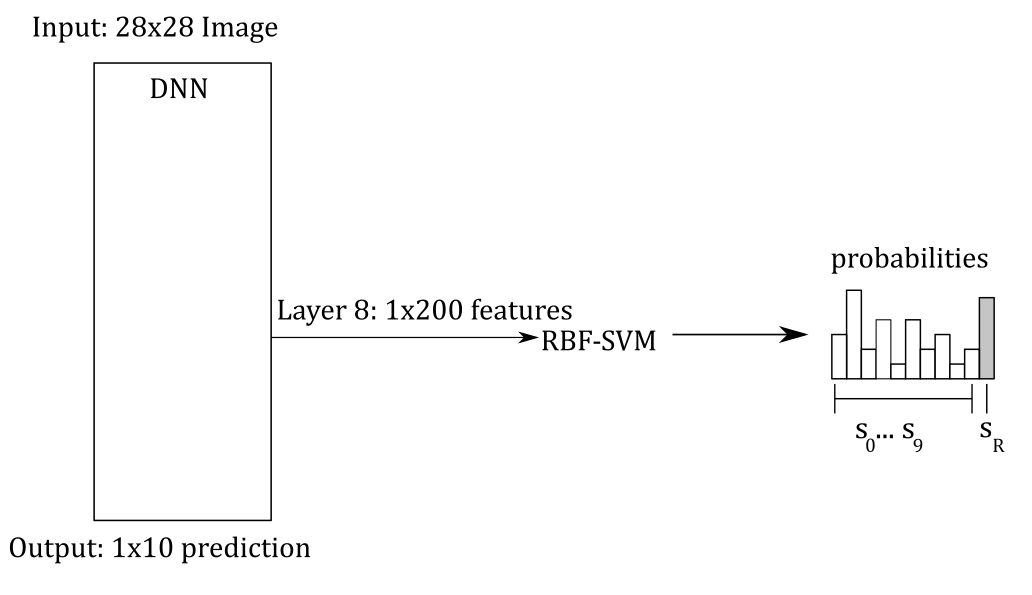


Figure – NR Defence Architecture

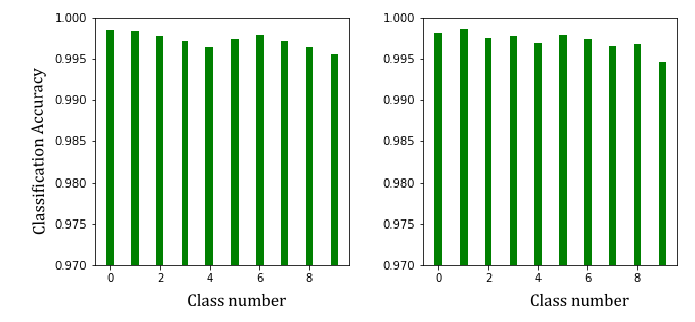


Figure – Training (left) and Testing (right) Classification accuracy

# Deep neural rejection (DNR)

Due to the effectiveness at reducing the effectiveness of evasion attacks found by the defence proposed by Melis et.al, the combined use of multiple layer’s features to better recognise perturbations was proposed by [11] as illustrated by Figure 5. The training of each SVM was completed similarly to that of NR however the features from layer 5 needed to first be flattened into a vector prior to being input into its respective SVM. Predictions from these SVM models were then concatenated into a 1x30 vector which was consequently fed into the final SVM model for training. Similar to previous, each model was trained individually of each other. Note once again the rejection mechanism implemented in the same manner as in the NR architecture, but only in the final SVM.

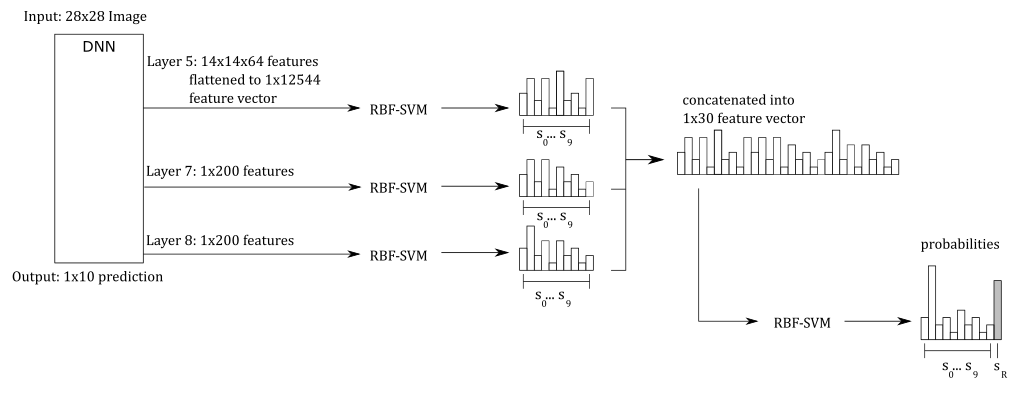


Figure – DNR Defence Architecture

The accuracies achieved by each of the 3 SVMs (‘sub-SVMs’ trained directly on the extracted features can be seen illustrated in Figure 6. The accuracy achieved by the final concatenated SVM can be seen illustrated in FIGURE.

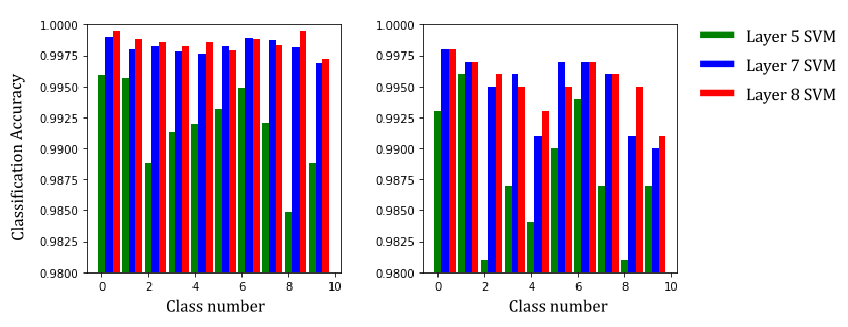


Figure – Training (left) and Testing (right) Classification accuracy for each individual ‘sub-SVMs’

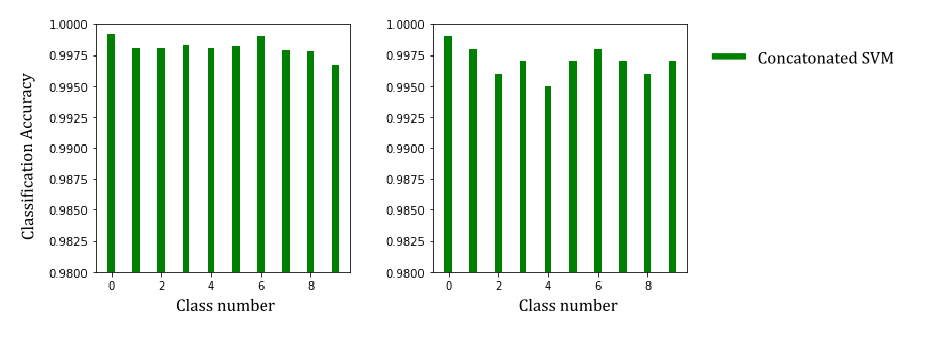


Figure – Training (left) and Testing (right) Classification accuracy of final concatenated SVM

# adversarial learning

As seen implemented by such defences as REFERENCE, the use of adversarial learning is the other commonly implemented strategy in the mitigation of successful adversarial attacks. This is on the basis that adversarial images exist due to a lack of regularisation of the model as suggested by [REF] and exploit ‘pockets’ of untrained space within a classifier. Adversarial learning aims to retrain the classifier using generated adversarial examples thus ‘filling’ these pockets of space lacking training, consequently improving the generalisation of the model.

This was completed as a two-step process where the original classifier was similarly attacked to the models above, and the successful adversarial examples used to retrain the original model. This original model was then re-attacked to observe the effectiveness of the consequent attack as illustrated by Figure 6.

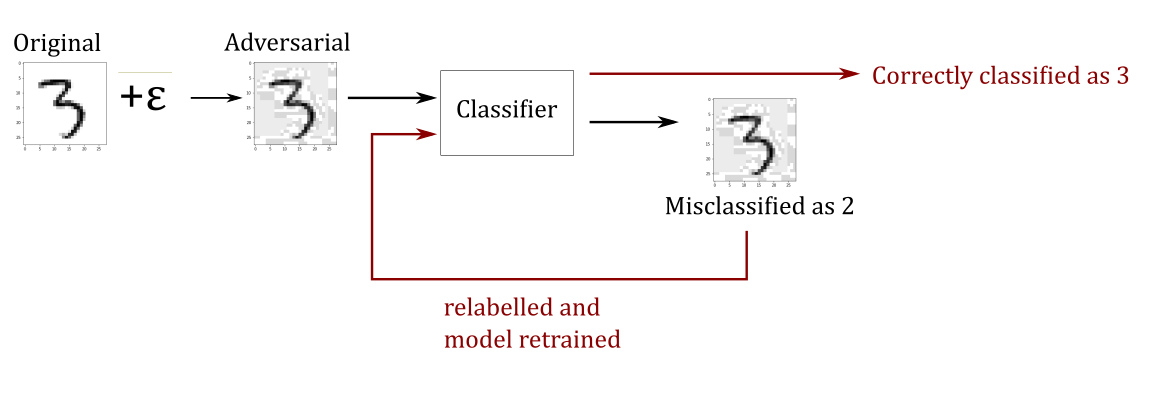


Figure – Adversarial Training Process

# Attack methods

An attack on a deep neural network is directly dependant on the level of the attacker’s knowledge of said model. They can essentially be separated into two general categories:

* A PK attack – a scenario where the adversary has perfect knowledge of the targeted classifier. As a result, the attacker can be assumed to have access to all model parameters such as its gradients, but is unable to physically change the model and does not have access to the training data.
* A LK attack – a scenario where the attacker has a limited knowledge of the targeted classifier.

Intuitively, the more knowledge that an adversary has of the model, the greater effectiveness of the attack as has been demonstrated by multiple experiments. [REFERENCE]. For the purpose of thoroughly testing defence techniques, a ‘defence-aware’ gradient attack will be implemented where the attacker will have a perfect knowledge (PK) of our defence architecture.

# Evasion attack method

The adversary’s goal is to take any sample x0 and through an attack strategy find a sample x\*, where:

This amounted to solving the following constrained optimisation problem:

Note, that max(sj) is the maximum of any of the other not-true class scores. This naturally leads the nature of the attack to being untargeted. This can be adapted to perform a targeted attack if needed; however, to solely test the robustness of a defence, this was decided unnecessary.

Through an iterative process, this optimisation to minimise the score of the true class and maximise a false class can be solved by making small perturbations to the input image. A successful attack will therefore be one which is able to successfully manipulate an input image, whilst being within the L2 boundary (ε), to cause a misclassification. 1000 random images were taken

# Projected Gradient descent (pgd)

It was possible to implement this evasion attack through a process of projected gradient descent (PGD).

Given an example image (x0) with label y0 and probability s0 as such:

Its adversarial target can be obtained (the second highest probability prediction) and consequently an adversarial label made to match this target. In the example mentioned, it can be seen that for x0with y0⇒3, the second highest prediction is for class 1. Hence an adversarial label, y\*, can be made as such for the intended adversarial example x\*:

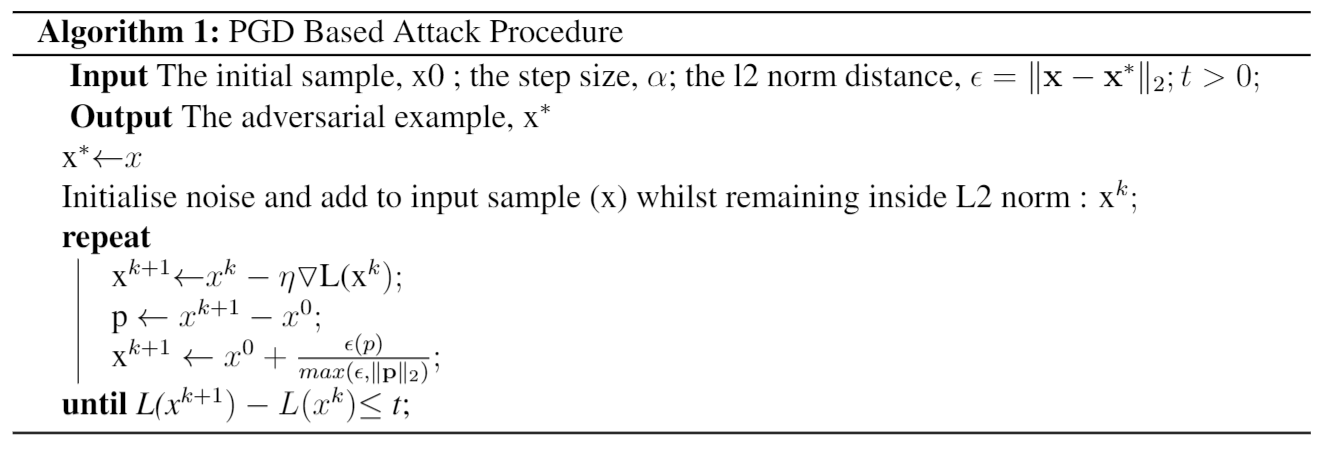
Hinge loss is typically used for multiclass classification applications, and is notably used in support vector machines. It can be detailed as such:

For the purpose of creating an adversarial example, this loss function instead made use of y\* in place of yi for x\* with the objective of reducing the loss for the adversarial label. This can be detailed as such:

Through gradient descent, this loss could then be minimised by applying small perturbations to the input image through an iterative process as such:

If xk+1 exceeds the l2-norm distance (ε) that is predefined by the utilised constraint, it is projected back inside as so:

This process of creating an adversarial example can be more succinctly described by Algorithm 1.



Note that the constant ‘t’ was an arbitrarily chosen constant as to confirm a level of convergence to the algorithm. In the tests conducted, it was found sufficient for t=0.01.

The gradient of the loss function w.r.t to the input image was chosen to give an indicator of the direction of perturbation with the aim of reducing this loss.

The gradient of the Neural network portion of the architecture in question could be calculated simply using the Gradient Tape method provided by TensorFlow’s API: a method of automatic differentiation. Using techniques such as the chain rule, it was then possible to combine this with the gradient of the SVM portions of the architecture to get the overall gradient of the output loss w.r.t input image so that a perturbation could be applied. This gradient for the SVM portions could be calculated as such:

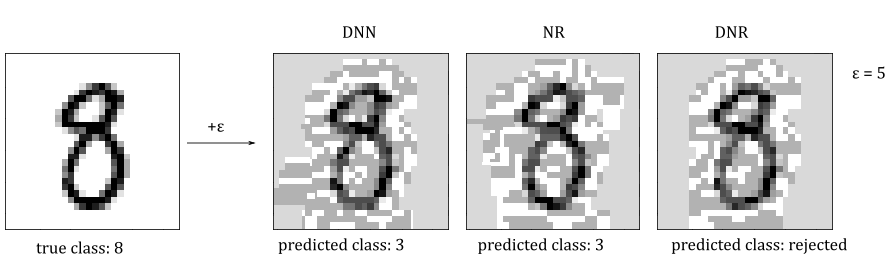
For the non-zero case, the gradient directly correlates to the gradient of the decision function.

We see that this gradient is then directly related to the gradient of the Kernel that was used. For an RBF Kernel, then:

# results

As mentioned in Section 2.1, the adversarial attack was used to create a dataset of adversarial examples. Examples of the adversarial images created for each defence method can be seen illustrated in FIGURE. This was then used as to extract the performance of each defence method. When considering the accuracy of the rejection based methods mentioned, a rejection of a sample under perturbation was considered a successful classification. For both the rate of rejection and accuracy of each architecture, see Figure 10.

The DNR method can be seen to perform to a similar degree up-to ε=3, at which point we can see the DNR defence method begin to out-perform. The rejection rate can be seen to follow a similar correlation, albeit not as severe. When performing a comparison between the research conducted by Sotgiu et al.[11], which this report tries to follow as closely as possible, and the results found here, a similar correlation between the three architectures can be seen. However, the attack conducted here, shows a far weaker response which can be a result of the attack itself not being as well optimized as possible. Some research have used extra components in addition to a loss function such as a density estimator. The use of such estimators penalise perturbations in low density regions and have been shown to favour attack points which imitate features of known samples such as those in mimicry based attacks [REFERENCES]. The nature of the adversarial images generated here seem to lack this element of density and hence result in a higher resolution of noise being perturbed in comparison to the far higher resolution noise elements that can be seen in the work conducted by Sotgiu et al. Given the difference in nature of the adversarial examples generated in this report and that of [11] where samples seem to be perturbed more precisely and can be observed to show the features of other samples, this may be an avenue to pursue in the future as to perform an even more severe attack. a



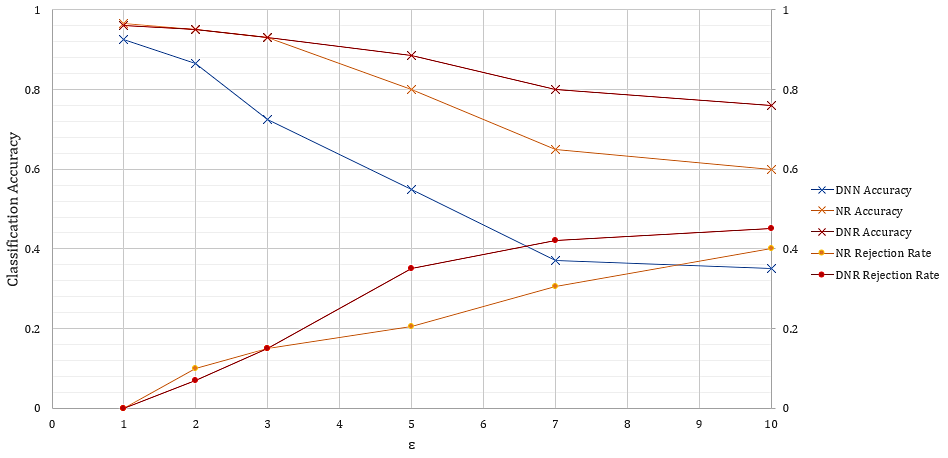
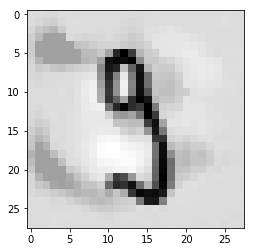
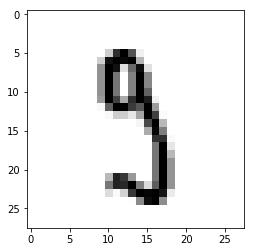


Figure 10 – Classification Accuracy and Rejection Rate results

Due to the good performance demonstrated by this DNR architecture, the techniques employed in adversarial training were consequently implemented on the defence. This was implemented as a means of understanding the effectiveness of combining two popular schools of thought involving the defence against adversarial examples. The base DNN was also retrained as to offer a better comparison between the methods.

When performing adversarial training, only the adversarial images which were able to cause a successful misclassification were used. Consequently, a newly created adversarial dataset was used to report the accuracy and rejection rate metrics seen in FIGURE.

# examples of adversarial images + adversarial features generated



Side by side comparison of typical images and then with the addition of adversarial noise

# analysis/discussion of results

# effectiveness of defence methods on speed and accuracy

# suggestion on the application of different defence methods

Safety criticality vs need for speed. Perhaps a method of defence can be something to do with how long an iteration takes? If an attacker is attempting to add something to the image this will take longer? MAYBE NOT]

# conclusion and future work

Discussion of the things I would have liked to also test but due to limitations in time I was not able to. Ways to possibly take this piece of work further

# references

1. V. Swaminathan, S. Arora, R. Bansal, R.Rajalakshmi. “Autonomous Driving System with Road Sign Recognition using Convolutional Neural Networks”, in *Second International Conference on Computational Intelligence in Data Science (ICCIDS-2019),* Chennai, India, 2019, pp. 1-4
2. G. Tiron and M. Poboroniuc, "Neural Network Based Traffic Sign Recognition for Autonomous Driving", in *2019 International Conference on Electromechanical and Energy Systems (SIELMEN)*, Craiova, Romania, 2019, pp. 1-5.
3. X. Wang, T. Qiu, C. Chen and N. Chen, "A Neural-Network-Based Real-end Collision Prediction Mechanism for Smart Cities", in *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, Tianjin, China, 2019, pp. 496-497.
4. C. Sowmiya and P. Sumitra, "Analytical study of heart disease diagnosis using classification techniques", in *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, Srivilliputhur, 2017, pp. 1-5.
5. A. S. Katasev and D. V. Kataseva, "Neural network diagnosis of anomalous network activity in telecommunication systems", in *2016 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, Omsk, 2016, pp. 1-4.
6. K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, “Robust Physical-World Attacks on Deep Learning Models”, arXiv preprint arXiv: 1707.08945, 2018.
7. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, “Intriguing properties of neural networks”, arXiv preprint arXiv:1312.6199, 2013.
8. N. Carlini and D. Wagner, "Audio Adversarial Examples: Targeted Attacks on Speech-to-Text," in *2018 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, 2018, pp. 1-7.
9. I. Goodfellow, J. Shlens, C. Szegedy, “Explaining and Harnessing Adversarial Examples”, arXiv preprint arXiv:1412.6572, 2014
10. M. Melis, A. Demontis, B. Biggio, G. Brown, G. Fumera, and F. Roli, “Is deep learning safe for robot vision? Adversarial examples against the iCub humanoid”, *in ICCVW Vision in Practice on Autonomous Robots (ViPAR),* pp. 751–759, IEEE, 2017.
11. A. Sotgiu, A. Demontis, M. Melis, B. Biggio, G. Fumera, X. Feng, F. Roli, “Deep Neural Rejection against Adversarial Examples”, arXiv preprint arXiv: 1910.00470, 2020
12. N. Carlini and D. A. Wagner, “Towards Evaluating the Robustness of Neural Networks”, arXiv preprint arXiv: 1608.04644, 2017
13. Keras FAQ: Frequently Asked Keras Questions [Online]. Available: https://keras.io/getting-started/faq/#why-is-the-training-loss-much-higher-than-the-testing-loss [Accessed: 17 April 2020]

# appendix

1. J.J. O'Connor and E.F. Robertson. *Gustav Robert Kirchhoff* [Online]. Available:   
   http://www-history.mcs.st-and.ac.uk/Biographies/Kirchhoff.html [Accessed: 13 Sept. 2013].

The proven effectiveness of machine learning and data-driven artificial intelligence in the area of pattern and object recognition has seen a rise in the subsequent implementation of such systems in many applications such as self-driving cars and malware detection software. Due to the safety and security criticality of such systems it is pertinent that they are made robust to withstand malicious attempts at disruption from adversarial examples. A deep neural network has been trained to classify the MNIST dataset. Evasion attacks have been implemented against this model and the effectiveness of said attacks against various defence methods that have been used.

SOME EXAMPLES OF ADVERSARIAL ATTACKS MISLEADING MODELS….REFER TO RESEARCH REPORT

AND EXAMPLE OF DEFENCE METHODS THAT HAVE ALREADY BEEN IMPLEMENTED

# IMPLEMENTATIONS/EXPERIMENTAL SETUPS

DISCUSS MY IMPLEMENTATION OF EVASION ATTACK IN DEPTH HERE (what libraries I used, snippets of pseudo code maybe. Include main code in appendix perhaps or instead a link to a github might be better than including in appendix). HOW DID I PERFORM PGD

DISCUSS MY IMPLEMENTATION OF POISONNING POSSIBLY

DISCUSS MY IMPLEMENTATION OF SINGLE LAYER NR HERE

DISCUSS MY IMPLEMENTATION OF DNR IN DEPTH HERE

DISCUSS MY IMPLEMENTATION OF USING A FILTER INSTEAD HERE…VARIOUS FILTERS USED. DISCUSS DIFFERENCE IN SPEEDS ADVANTAGES AND DISADVANTAGES

DISCUSS MY IMPLEMENTATION OF ADVERSARIAL LEARNING HERE

# EXPERIMENTAL RESULTS

TYPES OF IMAGES PRODCUED. DIFFERENCES IN SPEEDS AND ACCURACIES OF THE DIFFERENT METHODS

# ANALYSIS OF RESULTS

WHY MIGHT SOME SYSTEMS BE BETTER THAN OTHERS. POSSIBLE LIMITATIONS OF WHAT I HAVE DONE.

SUGGEST THAT SVMS HAVE BEEN PROVEN TO WORK VERY WELL ON MNIST IN PARTICULAR MIGHT NOT BE THE CASE FOR OTHER TYPES OF DATASETS. GIVEN THE TIMESCALE, IT WASN’T POSSIBLE TO ADDRESS THIS AS WELL AND SHOULD BE A PLAN FOR FUTURE WORK.

# CONCLUSIONS AND FUTURE WORK

WHAT DID I FIND OUT, FURTHER WORK, EXTENT TO WHICH MY AIM WAS ACHIEVED

MAYBE I DIDN’T GET TO FINISH POISONNING ATTACK

METHOD OF DISTILLATION AS A DEFENCE POSSIBLY TO BE TESTED IN THE FUTURE AND COMPARED AGAINST THE METHODS USED HERE

# Language

All text is to be written in UK English unless quoting from or referring to non-UK sources. Except in quotations the writing style should be impersonal, in the past tense and where possible gender neutral.

# Font

The default font is Cambria and Cambria Math for equations used in upright, *italic* and **upright bold** forms depending on the context. Times New Roman is an acceptable alternative where Cambria is not available.

# Page formatting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Paper size: A4, 210 mm × 297 mm | | | | |
| Margins: | Top: | 22 mm | Bottom: | 16 mm |
|  | Left: | 18 mm | Right: | 18 mm |
| Layout: | Header: | 10 mm | Footer: | 8 mm |

# Paragraph formatting

Normal paragraph text should be left-justified 10 pt upright and should extend over the full page width (174 mm). Line spacing should normally be 14 pt and paragraph spacing 6 pt after. Widow and Orphan lines should be avoided. A suitable paragraph style [Normal], is included in this document.

Word paragraph settings:

|  |  |  |  |
| --- | --- | --- | --- |
| Alignment: | Left | Outline level: | Body |
| Indentation, Left: | 0 mm |  |  |
| Right: | 0 mm | Special: | (none) |
| Spacing, Before: | 0 pt |  |  |
| After: | 6 pt | Line spacing: | At least 14 pt |

Under Line and Page Breaks, Widow/Orphan control should be on.

The title of the paper should be centre-justified 14 pt bold upright and should extend over the full page width but with left and right indentations of 8 mm. Line spacing should be 16 pt and paragraph spacing 12 pt after. A suitable paragraph style [Main-Title] is included in this document.

The Author’s name and ID number should be centre-justified 12 pt bold upright and should extend over the full page width but with left and right indentations of 8 mm. Line spacing should be 14 pt and paragraph spacing 6 pt after. Given and Family names should be ordered in the same way that they appear on the Author’s ID card. The name and ID number should be separated by a semi-colon. A suitable paragraph style [Author-Name] is included in this document.

The Author’s programme should be centre-justified 10 pt bold upright and should extend over the full page width but with left and right indentations of 8 mm. Line spacing should be 14 pt and paragraph spacing 12 pt after. A suitable paragraph style [Author-Programme] is included in this document.

The Abstract should be formatted as normal text but with left and right indentations of 8 mm and paragraph spacing 12 pt after. The paragraph should be identified with the in-line heading **Abstract:** in 10 pt bold upright. A suitable paragraph style [Abstract], is included in this document.

All headings should be left-justified 10 pt bold upright. Line spacing should be 14 pt and paragraph spacing 6 pt after. Headings should be kept with the following text (Under Line and Page Breaks, Keep with next should be on). Where required headings should be numbered sequentially in a 1, 2, 3 style with sub-sections in the same sequence, i.e. 1, 1.1, 1.1.1. No more than three sub-levels should be used. An unnumbered paragraph style [Heading] and three levels of automatically numbered paragraph styles [Heading-1, Heading-2 and Heading-3] are included in this document.

Figures should normally be centred on the page. All figures must be captioned and numbered sequentially in a separate series, in a 1, 2, 3 style and in the order they are referred to in the text. All figures must be referred to in the text and vice versa. Figure captions should be centred on the page and preferably in a single line. Figures with multiple sections should be labelled in an (a), (b), (c) style. An example of a suitably captioned full-width figure is shown below. It would be referred to in the text as Figure 1.

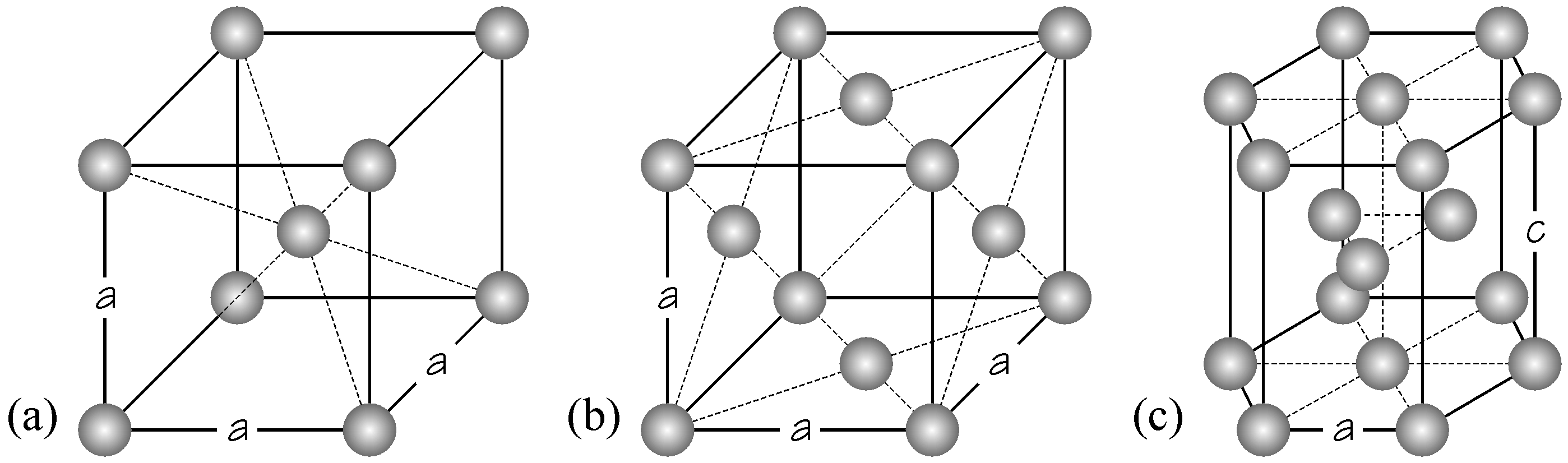


Figure 1 Example of a full-width multiple section figure [9]

A figure that is less than 90 mm in width may be located in a text-box, frame or single-cell table - the container - and text allowed to flow around it. The container should be left-aligned with text allowed to flow around the right-hand side. Box or cell spacing should be; Top: 0 mm, Right: 3 mm, Bottom: 0 mm, Left: 0 mm. The enclosed figure and caption should be formatted as for full-width figures and captions. An example of a suitably captioned half-width figure is shown on the left contained in a text box. It would be referred to in the text as Figure 2. Paragraph styles suitable for full- and half-width figures [Figure-1] and figure captions [Caption-1] are included in this document.

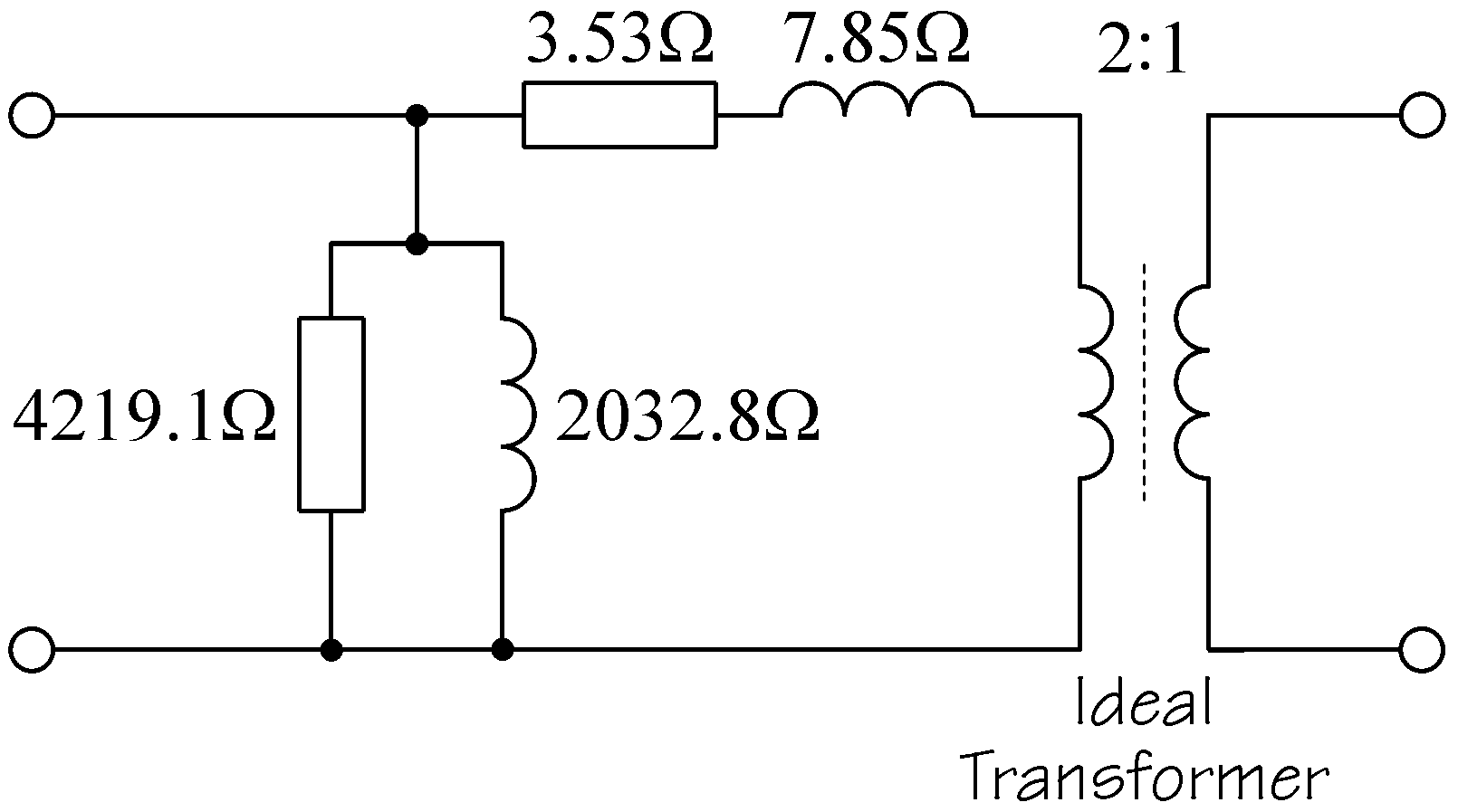


Figure 2 Example half-width figure

Using illustrations taken from the work of others must be justifiable, and their origin must be identified by including the source as a reference and adding a conventional citation to the caption, as illustrated in Figure 1.

Tables should normally be centred on the page. All tables must be captioned and numbered sequentially in a separate series, in a 1, 2, 3 style and in the order they are referred to in the text. All tables must be referred to in the text and vice versa. Table captions should be centred on the page and preferably in a single line. Text within tables should be set at single line spacing and with zero spacing before and after each paragraph. Cell spacing should be set at 1 mm. Tables should be formatted with main row separators only. An example of a suitably captioned full-width table is shown below. It would be referred to in the text as Table 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 |
| Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 |
| Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 |

Table 1 Example full-width table

A table that is less than 90 mm in width may be positioned on the left of the page and text allowed to flow around it.

|  |  |  |  |
| --- | --- | --- | --- |
| Column 1 | Column 2 | Column 3 | Column 4 |
| Data 1 | Data 2 | Data 3 | Data 4 |
| Data 1 | Data 2 | Data 3 | Data 4 |
| Table 1 Example half-width table | | | |

The table should be left-aligned with text allowed to flow around the right-hand side. The distance from surrounding text should be; Top: 0 mm, Right: 3 mm, Bottom: 0 mm, Left: 0 mm. The caption should be included in an additional row of the table. An example of a suitably captioned half-width table is shown on the left. It would be referred to in the text as Table 2. Paragraph styles suitable for full- and half-width table content [Table-1] and table captions [Caption-1] are included in this document.

Equations should be positioned in the normal style and left-indented by 10 mm. Equations should only be numbered when they are referred to in the text, un-numbered equations must follow in logical sequence from preceding text or equations. Equations should be numbered sequentially in a separate series in a 1, 2, 3 style and in the order they are referred to in the text. Numbers should be in round brackets and left-aligned at 160 mm.

Examples of suitably numbered equations are shown below. They would be referred to in the text as Equation 1 and Equation 2 respectively.

(1)

where: Ψ is the flux-linkage

*Nc* is the number of turns in the coil

Φ is the useful flux/pole

or

where: Ψ is the flux-linkage, *Nc* is the number of turns in the coil and Φ is the useful flux/pole

**** (2)

A suitable paragraph style for equations [Equation-1] is included in this document.

Note that equation numbering in this form can cause problems in Microsoft Word when using the built-in equation editor (not the MathType clone). An equation in the same line as text is often automatically re-formatted. This is most easily overcome by putting the equation and its number in a one row, two column table, for example;

|  |  |
| --- | --- |
|  | (1) |

The table column widths should be 160 mm and 14 mm with the equation number left-aligned in the second column.

# Text formatting

* Full stops should be followed by a single space.
* All Figure, Table and Equation designations should be capitalised and separated from their associated number by a non-breaking space, i.e. the designation and number should be kept together.
* Numbers should be separated from their units by a non-breaking space, i.e. numbers and units should be kept together.
* Normal text in figures and tables should be no smaller than 8 pt.
* Vectors and matrices should be in 10 pt bold italic.
* Variables should be in 10 pt italic. Italics should not be used for emphasis.
* Functions and operators should be in 10 pt upright.
* Numeric indices should be in 7 pt upright.
* Letter indices should normally be lower-case 7 pt italic.
* All upper-case Greek letters should be in upright text.
* A multiplication sign (×) should be used to indicate multiplication of numbers and numerical values, including values with units such as 5 mm × 4 mm. The letter “x” or a centre dot “·” should be avoided (unless representing a dot product).
* The decimal point should be represented by a dot on the line, i.e. a full stop, not a centre dot or comma.
* Exponential expressions should be written in superscript form, i.e. 12 × 103 not 12E03.
* Upright or stacked fractions are normally preferred but single line variations using the solidus, such as x/y, are acceptable where necessary or appropriate, e.g. in text.
* Acronyms and abbreviations should be clearly defined the first time they are used in the text by writing the term in full followed by the acronym/abbreviation in round brackets.
* All dates should be written in full to avoid day-month confusion, e.g. August 29th 2005.

Note that a suitable paragraph style for the inset and bulleted paragraph used here [Inset-1] is included in this document.

# Computer code

Short sections of computer code can be included in the main text where absolutely necessary in the explanation of an algorithm or process otherwise code should be in a separate Appendix section [see Appendices and supplementary material]. Code should be set apart from conventional text by formatting it in a monospaced font such as Courier.

# Footnotes and References

If it is absolutely necessary to add additional comment that would disrupt the flow of text then a footnote can be used. Footnote markers in the text should be superscript Arabic numerals in 7 pt upright. Footnote text should be in left-justified 9 pt upright, single line spaced with 0 pt after.

References should be numbered sequentially (following the IEEE referencing format) in a separate series in a 1, 2, 3 style and in the order that they are first cited in the text. Citations can be positioned directly following the relevant phrase or at the end of a sentence or paragraph, but always before the full stop. Citations are made with full-size, in-line (i.e. not superscript) numbers enclosed in square brackets, e.g. [1]. Reference numbers in multiple citations should be separated by commas without spaces, [1,2], or a range can be indicated with a hyphen without spaces, [1–3]. References can be cited as many times as necessary, i.e. they should not be duplicated if they are cited more than once. If it is necessary or appropriate to cite specific pages in a reference that is cited more than once then the relevant page number(s) should be included in the citation as, for example, [5, p17] for a single page or [5, pp45-85] for a range of pages. Note that it is not unusual for page numbering to be different in different editions of a book so the edition should be identified in the reference where possible. All references should be cited in the text. References are an integral part of any paper and should not be relegated to an appendix.

References should provide as much information as possible to allow the reader to locate the material concerned. Typically references should contain; all author name(s) and initials, the title of the paper, date published, the title of journal or book, the volume or edition number, editors (if any), and finally the page range. For books and conferences the publisher and town of publication (if known) should also be given. For internet references as much information as possible about the authorship of the material and the date that the pages were last accessed should be included. Example references are shown below.

Journal paper:

1. G. Bertotti. “General properties of power losses in soft ferromagnetic materials”, *IEEE Transactions on Magnetics*, vol. 24, pp. 621‐630, Jan. 1998.

Conference paper:

1. J.G. Kettleborough, I.R. Smith, L.T.M. Fernando and B.A. Fanthome. "Numerical solution of electrical power systems using diakoptics", in *4th International Conference on Mathematical Modelling*, Zurich, Switzerland, 1983, pp. 1-6.

Book:

1. P.P. Silvester and R.L. Ferrari. *Finite elements for electrical engineers*, 3rd ed. Cambridge: Cambridge University Press, 1996.

Book chapter:

1. S.W. Director. “Analysis of networks containing resistors and current sources”, in *Circuit theory, a computational approach*, S.W. Director, Ed., New York: Wiley, 1975, pp. 45-85.

Thesis:

1. C.R. Fitton. "Mathematical modelling of balanced and unbalanced HVDC power transmission links", PhD thesis, Loughborough Univ. of Tech., 1988.

Standard:

1. *General requirements for rotating electrical machines*, BS 4999-147, 1988, IEC 60136, 1986.

Web site:

1. J.J. O'Connor and E.F. Robertson. *Gustav Robert Kirchhoff* [Online]. Available:   
   http://www-history.mcs.st-and.ac.uk/Biographies/Kirchhoff.html [Accessed: 13 Sept. 2013].

Previous project report or module lecture notes:

1. A.N. Other. "Project title," Dept. of Elect. and Elec. Eng., Loughborough Univ., Project Rep., Nov. 2009.
2. K. Gregory. "A note on the resistive loss in a winding," 12ELB003, School of Elect., Elec. and Sys. Eng., Loughborough Univ., 2012.

All references should be grouped together in the last section of the paper and listed sequentially and continuously. The reference paragraph should be left-justified 10 pt upright, should extend over the full page width, have a left hanging indentation of 8 mm, at least 14 pt line spacing and paragraph spacing 0 pt after. A suitable automatically numbered paragraph style [Ref-1] is included in this document.

Note that the explanatory headings (e.g. Book chapter) used above should not be included in the references. For example, a reference section including all of the example sources should appear as;

1. G. Bertotti. “General properties of power losses in soft ferromagnetic materials,” *IEEE Transactions on Magnetics*, vol. 24, pp. 621‐630, Jan. 1998.
2. J.G. Kettleborough, I.R. Smith, L.T.M. Fernando and B.A. Fanthome. "Numerical solution of electrical power systems using diakoptics", in *4th International Conference on Mathematical Modelling*, Zurich, Switzerland, 1983, pp. 1-6.
3. P.P. Silvester and R.L. Ferrari. *Finite elements for electrical engineers*, 3rd ed. Cambridge: Cambridge University Press, 1996.
4. S.W. Director, “Analysis of Networks containing resistors and current sources, ” in *Circuit theory, a computational approach*, S.W. Director, Ed., New York: Wiley, 1975, pp. 45-85.
5. C.R. Fitton: "Mathematical modelling of balanced and unbalanced HVDC power transmission links", PhD thesis, Loughborough Univ. of Tech., 1988.
6. *General requirements for rotating electrical machines*, BS 4999-147, 1988, IEC 60136, 1986.
7. J.J. O'Connor and E.F. Robertson, *Gustav Robert Kirchhoff* [Online]. Available:   
   http://www-history.mcs.st-and.ac.uk/Biographies/Kirchhoff.html [Accessed: 13 Sept. 2013].
8. A.N. Other, "Project title," Dept. Elect. and Elec. Eng., Loughborough Univ., Project Rep., Nov. 2009.
9. K. Gregory, "A note on the resistive loss in a winding," 12ELB003, School of Elect., Elec. and Sys. Eng., Loughborough Univ., 2012.

# References or bibliography?

There is often confusion about the difference between references and a bibliography. A reference is made to a specific source that has been "referred to" or "cited" explicitly in the text. A citation means "You should look at this it is important" or "This material supports or contradicts my own" or "I used this material or based elements of my work upon it". A bibliography is simply a list of sources that the author has read that may be relevant to the subject but that have not been explicitly cited in the text. In general, there is no need for a bibliography in a technical paper, if a source is worth referring to then it should be cited explicitly. In addition, any relevant bibliographic material would probably have formed part of the research report.

# Using the research report

Material from the preceding research report can be included in the final paper, although this is essentially self-plagiarism in that marks would be awarded twice for the same material. Since duplication would be detected by the text matching software by which all research reports and final papers are vetted, and excessive duplication would not be looked on favourably, it is better to simply refer to the research report in the usual way, i.e.

1. A.N. Other, "Project title," Dept. Elec. Elect. Eng., Loughborough Univ., Project Rep., Nov. 2009.

# Acknowledgements

Acknowledgements should be in a separate section between the Conclusions and References.

# Appendices and supplementary material

Additional material, e.g. mathematical derivations that may interrupt the flow of an argument, should form a separate appendix section. Appendices should not be used to lengthen the paper unnecessarily; if the material can be found in another source then this should be cited as a reference not reproduced.

Appendices should be numbered sequentially in a separate series in a 1, 2, 3 style and in the order that they are referred to in the text. Appendix designations should be capitalised and separated from their associated number by a non-breaking space, i.e. the designation and number should be kept together. All appendices must be cited in the text with their designation and number in square brackets, e.g. [Appendix 1].

# Page limits

The main text should be no more than 12 pages (sides of A4) and an additional 12 pages (sides of A4) of appendices may be added. The 12 (+ 12) page limit is inclusive; i.e. it includes *everything* that is deemed to be required.

– & Chris Ward – Revision 4 – 28 April 2020