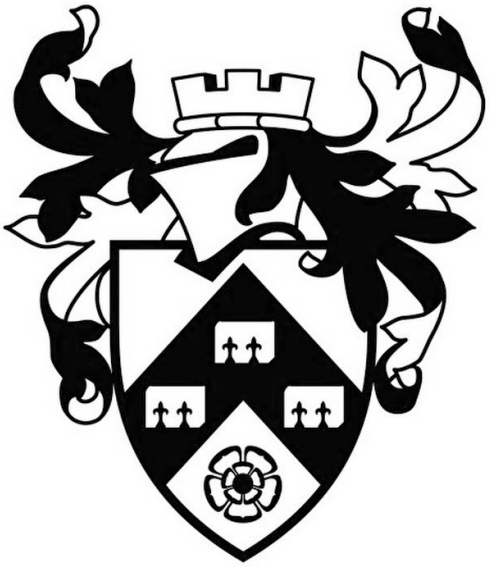
Using Neural Network Interpretability Methods to Understand Knowledge Distillation



John Gardner

A thesis presented for the degree of

Master of Computer Science

University of York, February, 2022

Acknowledgements

First and foremost, I am very grateful for the help, guidance and support of my supervisor while undertaking and writing up this research. Thank you Ojie!

I would also like to thank my family for their support during the entirety of this Masters program, and my friends for ensuring that I enjoyed the rest of life away from it.

Executive Summary

The work presented in this report aims to experimentally ratify a recently proposed theory that attempts to explain the mechanism behind Knowledge Distillation (KD) in Deep Learning.

KD is a method that can significantly improve the accuracy of a single Neural Network (NN) model, and finds use in applications with strict test-time energy and latency requirements. However, this technique comes at the high computational cost of having to train several precursor NN models. Many techniques have therefore been developed that aim to deliver the same performance gain as KD without the increased training overhead. However, none of these can deliver the same degree of performance increase.

The motivation for this work is therefore to build upon this new theory and improve intuition and understanding about the mechanism of KD. This in turn can then inform the development of new and improved techniques that mimic KD without the computational overhead.

To meet this goal, KD has been performed by training a collection of Deep Convolutional NNs on a popular image set. An occlusion-based input attribution technique has then been used to generate saliency maps for a range of test-set images, highlighting the features that these NNs have learned.

Subsequent analysis of the distribution and evolution of these feature sets shows that, using modern techniques, 25 Epochs of training is sufficient for a robust set of features to be learned by a deep NN, that the feature sets learned by different NNs vary widely when trained on image data, and that the feature set of a distilled network closely match those of the ensemble it is distilled from. All these results ratify the theory mentioned above.

The dataset used for this research is publically available, contains no personal information, has been used solely for the purpose of this research and has been discarded after use. No other ethical, legal, social, or professional have presented themselves in this work.

Contents

Glossary and Acronyms

Term	Meaning
Knowledge Distillation	The process of transferring the knowledge and representations learned by a model into a smaller one, giving rise to a similar level of performance at test time.
CIFAR10	A common image classification dataset, containing 10,000 32px x 32px images for each of 10 classes of everyday objects: *plane, car, bird, cat, deer, dog, frog, horse, ship* and *truck*.
multi-view data	Data on which classification is carried out, and in which every class has many characteristic features or views associated with it. Each data instance may only have a subset of these views present. Image data is a prime example of this data type.
multi-view hypothesis	The hypothesis that the training of Deep Neural Networks on multi-view data is comprised of distinct two phases: phase-one and phase-two (see below).
phase-one	The first phase of training, where Deep Neural Networks are performing hierarchical feature learning. This phase continues until a significant majority of data instances can be classified from the subset of features the model has learned.
phase-two	The second phase of training, where the model has stopped learning new and general feataures, and instead memorises the remaining training instances that are missclassified.

Acronym	Expanded
(D)NN	(Deep) Neural Network
KD	Knowledge Distillation
SGD	Stochastic Gradient Descent
RICAP	Random Image Cropping And Patching

Chapter 1

Introduction

1.1 Background

Deep Neural Networks are a powerful class of models that can approximate any function to arbitrary precision given sufficient parameterisation [1]. In the modern world, they find use in domains and tasks as varied as medical data analysis [2], drug discovery [3, 4], sentiment analysis [5], self-driving cars [6, 7] and many more [8]. In particular, the ability of Deep NNs to learn hierarchical sets of features makes them well suited to the task of image classification [4, 9].

Techniques are constantly being developed to increase the performance of Deep NNs, improving their already substantial utility to humanity [8]. One such technique, first proposed in 2015, is Knowledge Distillation [10]. In summary, this involves training several DNNs independently on the same dataset, forming an ensemble from the unweighted mean of these DNNs' outputs, and training a further DNN (typically with identical architecture to the originals) to mimic the output of this ensemble on the training set, rather than on the data's original labels. In many cases and domains, ensembling leads to a large test-time performance increase, but always at the cost of increased latency and energy usage [10, 11]. Subsequent Knowledge Distillation produces models that retain much of the ensemble's performance increase without these additional inference costs. This technique is therefore crucial in settings with tight latency or energy requirements.

The origin of the performance increase seen both when ensembling Deep NNs and when performing Knowledge Distillation remains debated [10, 11, 12, 13, 14, 15]. In a recent (2021) paper *Allen-Zhu and Li* attempt to explain both phenomena within the context of their newly proposed "multi-view" data paradigm [11]. The "multi-view" structure refers to data where many "views" or features are associated with a class, but where only a subset of these are associated with any one training instance. Images are a prime example of this common data paradigm - to illustrate this,

Allen-Zhu and Li provide the following example: pictures of cars typically contain such obvious features as headlights, tyres, windscreens and doors. Many pictures will have all of these features present. However, in pictures taken from odd angles, or when parts of the car are occluded, some of these features can be missing. Despite this, it is still easily possible to identify the car from the subset of views present.

Allen-Zhu and Li propose that the training of Deep NNs on such multi-view data can be split into two phases [11]: in the first phase, the network quickly learns a subset of views/features that suffice to classify most training instances. Once these features have been learned, the correctly identified instances contribute negligibly to the gradient updates during the rest of training. This leads to the second phase, in which the network merely memorises (i.e. overfits to) the remaining training data that are incorrectly classified. These phases are subsequently referred to as phase-one and phase-two respectively.

From this proposition, it follows that ensembling amounts to creating a model with a wider set of incorporated features whose output is less affected by signals originating from the spurious overfitting in phase-two. Thus, the proportion of test-data that can be identified from reliable features is higher for the ensemble than for any of its component models, meaning that the ensemble displays better generalisation properties.

Allen-Zhu and Li's multi-view theory also supports the Dark Knowledge theory of Knowledge Distillation [15]: the output of the ensemble contains semantically richer information than the hard targets present in the original data, and hence training to mimic the ensemble's output forces the distilled NN to learn a broader set of views/features than the original NNs.

1.2 Motivation

The original Knowledge Distillation procedure described above finds applications wherever Neural Networks are used, and in particular in the fields of computer vision and natural language processing, where deployed networks are required to be lightweight yet highly performant [16, 17].

Many other procedures have since been developed to also perform knowledge distillation, including self-distillation [11, 18], lifelong learning [19], and adversarial KD [20]. However, all these techniques come at a significant cost, since training (larger) teacher models requires additional time and resources [16, 17]. Several techniques therefore exist that aim to deliver the same performance boost as teacher-student KD without the need for additional training.

One such technique is the use of label-smoothing [21] (Figure 1.1). This involves replacing one-hot encoded labels, where each image corresponds to exactly one class, with "smoothed"

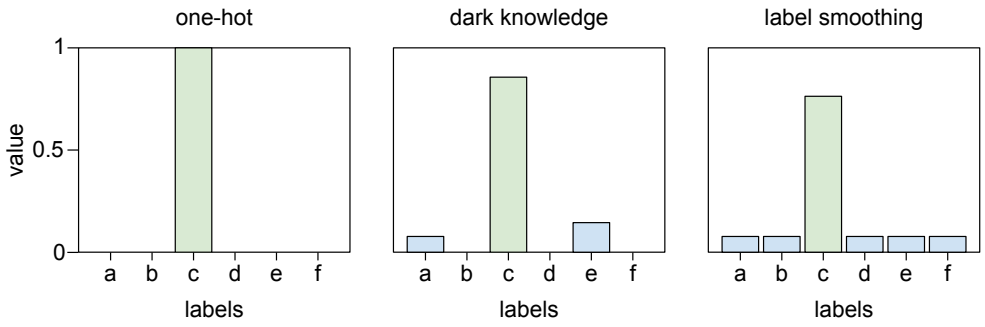


Figure 1.1: A comparison of different labeling methods.
Images are typically labeled with one-hot encodings, corresponding to a single class ('c' above). The Dark Knowledge present in the output of an ensemble suggests this image also has some 'a' and 'e' character. One interpretation of label smoothing is mimicing this Dark Knowledge without first training an ensemble.

labels, where each image is labeled with non-0 probability for all classes. The motivation behind this technique is to mimic the Dark Knowledge present in the output of the teacher ensemble model without having to train it. This label-smoothing can also be interpreted as a form of regularization, preventing overfitting and therefore improving test-time accuracy [22]. This leads to significant performance increases, but does not approach the power of full KD [21]. An explanation for this is that the label-smoothing performed by the ensemble is more meaningful due to the dark-knowledge present in its output [16, 21]. Additionally, the increase in performance from this smoothing quickly vanishes as the number of classes increases. This is undesirable in the computer vision context, where datasets often contain 100's or even 1000's of different labels [23, 24].

Another such technique is data augmentation using Random Image Cropping and Patching (RICAP) [25]. This involves using different versions of an image for each epoch of training. The removal of various parts of the image, either by cropping or patching, will naturally sometimes lead to the removal of key features in the image, and so this technique forces a NN to learn a more varied set of features, in turn leading to better generalisation capabilities. Again, however, full KD leads to a larger increase in performance than RICAP [11, 16, 25].

Given that none of these "teacher-less" techniques can provide the same performance benefit as the more costly full KD, *Allen-Zhu and Li's* motivation for the development and exposition of their multi-view theory was to

> "help design new, principled approaches to improve the test accuracy of a neural network, potentially matching that of ensemble"

by improving the theoretical understanding of the KD process [11].

However, *Allen-Zhu and Li* only prove their multi-view theory for the limited case of a 2-layer NN trained on a toy dataset with artificially multi-view nature. It remains to be shown that the multi-view theory indeed holds true for more realistic cases, including the training of DNN classifiers on image data. Additionally, accepted studies in Psychology show that the application of abstract and theoretical results to everyday scenarios is important in building intuition and absorbing information [26, 27]. Therefore, an exploration of the application of the multi-view theory to the common task of image classification (currently missing from the literature) will be beneficial to the research community, both to ratify that the theory holds more generally, and to improve intuitions concerning NN training that *Allen-Zhu and Li* have aimed to build.

Exploring how quickly the feature learning phase ends during training on real world data may be useful in developing new early stopping and feature regularization strategies. Additionally, exploring how multi-view real-world data actually is, and the types and number of features that different NNs can learn from them, may be useful in helping to accelerate the development of new and improved data augmentation strategies.

1.3 Overview and Research Questions

The work presented in this report seeks to experimentally ratify and explore the implications of *Allen-Zhu and Li's* multi-view theory for the case of a deep residual NN trained on a standard image dataset [9, 23]. Concretely, the following Research Questions are considered within this context:

RQ1: **How quickly** does phase-one, the learning of the final set of features, complete during the training of a Deep NN?

Justification: Allen-Zhu and Li only qualify that the first phase of training, where feature learning occurs, happens "quickly". The dynamics of the NN training process are known to be a function of the dataset, NN architecture and the training regime [13, 28]. However, by using standard techniques and architectures, the answer to this question will be broadly applicable to many other image classification tasks. Additionally, developing a way to measure the onset of phase-two (memorisation) during training will be particularly useful in creating new NN training approaches that combat overfitting through the early stopping process [11, 28].

RQ2: **How much** do the feature sets learned by different Neural Networks vary, and **of what**

quality is this variation?

Justification: The toy example considered by *Allen-Zhu and Li* has an artificially engineered multi-view nature. If real life image data is multi-view, as they claim, then different NNs should learn very different sets of features. As *Allen-Zhu and Li* point out, forcing models to learn multiple views directly from the data, rather than using the costly Knowledge Distillation process, would be a key goal of future techniques based on the multi-view hypothesis. To do this, understanding what exactly the views in real data are, and whether they correlate with those a human might assign, is a crucial first step. Subsequent research can then focus on developing more principled data-augmentation techniques [11].

RQ3: Compared to the variation above, **how similar** are the learned feature sets of an ensemble and a NN distilled from it?

Justification: If *Allen-Zhu and Li's* theory holds, the feature set of a distilled DNN should be closer to that of the ensemble than any of the original models are. The similarity between the feature sets of the ensemble and distilled net is important. In particular, exploring whether the distilled net learns *all* features present in the ensemble, or merely a subset is unexplored by *Allen-Zhu and Li*, yet is important to ascertain with a view to creating new and more effective techniques for NN training.

1.4 Aims

The aim of this research is to experimentally build upon *Allen-Zhu and Li's* multi-view theory, for the specific task of image classification, by seeking to quantify the rate at which feature learning takes place, to quantify and qualify the difference between feature sets learned by different NNs, and to confirm that the feature sets of distilled networks mimic those of the ensemble from which they are trained.

1.4.1 Objectives

Meeting these aims requires the completion of several objectives:

Obj. 1: Perform a review of the relevant literature pertaining to KD and to feature analysis.

Obj. 2: Identify the challenges of using these techniques given the limited scope of this work.

Obj. 3: Train a set of Deep NNs on the task of image classification.

Obj. 4: Develop and implement methods to perform *comparative* feature analysis of DNNs.

Obj. 5: Use these methods to answer the RQs posed above.

1.5 Research Scope

Allen-Zhu and Li's claim that their multi-view hypothesis holds for the training of Deep Neural Networks on a wide set of domains, both in terms of model architectures and of the type of data (images, text etc.) [11]. The research presented in this report has been carried out under limited time and budget constraints. To ensure maximum impact, a small subspace of the deep learning domain has therefore been selected: residual networks trained on a popular image dataset [9, 23]. This follows on from the little empirical analysis performed by *Allen-Zhu and Li* by using the same architecture and dataset. By focussing on this smaller setting, a conclusive study can be performed, providing much deeper insight than a shallow study performed over many types of model and data.

1.6 Structure of This Report

The structure for the remainder of this report is as follows:

Chapter 2 presents a literature review, in which the motivation for this project, the techniques used and the derivation of the Research Questions are further justified. Analysis of the relevant literature pertaining to both Knowledge Distillation and NN Interpretation are also put forward.

Chapter 3 presents the novel methodology developed as part of this work. The training regimes and analysis techniques used are described in sufficient detail to be completely duplicated (code is also made available to replicate all research presented in this report).

Chapter 4 presents an analysis of the findings of this research.

Chapter 5 presents conclusions of the undertaken research, together with limitations and suggestions for future directions of enquiry.

Chapter 2

Literature Review

The literature surrounding both Knowledge Distillation and Neural Network Interpretation, while extensive, has some obvious omissions. In this chapter, a review of the literature is put forward that attempts to summarise the current state in the field, and to highlight the new direction that the research presented in Chapters 3 and 4 hopes to take, namely that of a comparative feature analysis to probe *Allen-Zhu and Li's* multi-view theory in a real world context.

2.1 Deep Ensembles and Knowledge Distillation

Ensembling methods are a general and powerful set of techniques for improving test-set accuracy over a wide range of model classes [29, 30]. The resulting performance increase is well understood for many of these techniques, including boosting [31, 32], bagging [33], bootstrapping [30], and combining models of different architectures [34].

Authors	**Year**	**Paper**	**Theory**
Brown, Wyatt and Tiño	2005	Managing Diversity in Regression Ensembles	Ensembling leads to error cancellation
Allen-Zhu, Li and Song	2018	A Convergence Theory for Deep Learning via Over-Parameterization	Deep NNs operate within the Neural Tangent Kernel Regime. Ensembling thus leads to a larger set of random features, and therefore better generalisation.
Allen-Zhu and Li	2021	Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning	Ensembling on multi-view data works very differently (see section)

Table 2.1: competing theories for the origin of the performance increase seen when ensembling Deep NNs.

Despite this, consensus on the origin of improved performance for ensembles of *Deep NNs* remains lacking in the literature: none of the above theories can justify the case where identical models are trained on the same dataset and differ only in the randomness seen during initialisation and training. Other competing theories exist which aim to explain this phenomenon (Table 2.1). As discussed by *Gou et al*, however, these are limited in number [16]:

> "there are not too many works on either the theoretical or empirical understanding of knowledge distillation".

Brown, Wyatt and Tiño, along with others, propose that NNs learn some function $F_i(x) = y + \varepsilon_i$ that approximates the true function y together with some noise ε_i. Ensembling over several such models amounts to reducing the impact of this noise: $\sum_i \varepsilon_i < \varepsilon$, assuming that these noise values are independent [12, 14]. This in turn leads to less noisy, and therefore more accurate predictions [12, 14].

Previously, *Allen-Zhu and Li* have argued that ensembling of deep NNs works analogously to ensembling in the Neural Tangent Kernel regime, where NNs do not perform hierarchical feature learning, and instead learn a linear function of the random features that are determined by random initialisation. Ensembling in this case improves performance merely because a larger set of random features is used. [11, 35, 36, 37, 38].

More recently, *Allen-Zhu and Li* show that neither of these common explanations is correct or relevant to the now standard context of deep, narrow, over-parameterised NNs trained using Stochastic Gradient Descent (SGD) with momentum on multi-view data [11]. Instead, they propose the new "multi-view" hypothesis detailed in Chapter 1. They prove this theory for a specialised and engineered toy example, but do not explicitly demonstrate that this generalises to every day contexts, such as deep image classification.

The multi-view hypothesis also explains the mechanism by which the Dark Knowledge present in ensemble outputs leads to the higher performances of distilled networks. If true, this hypothesis additionally supports the less rigorous explanation of this phenomenon provided by Hinton et al in their original KD paper [10].

The new theory proposed by *Allen-Zhu and Li* provides deep insight into the training of NNs. Confirming that it holds in more general settings, such as image classification using standard architectures, rather than just for a toy system, is therefore crucial if these insights are to be incorporated into new and advanced NN training techniques. The lack of any studies focussing on this in the literature thus provides the motivation for undertaking this work.

2.2 Neural Network Interpretation

The research presented in this report seeks to compare the feature sets learned by different NNs. A review of the techniques developed to interpret deep NNs is therefore included below.

Deep NNs can represent arbitrarily complicated decision boundaries and this "black box" nature makes them difficult to interpret [39]. However, many applications of NNs require high confidence in their outputs, and therefore much research effort is placed on developing techniques to improve NN explainability [40].

Of relevance to this research are the family of interpretability techniques used to determine and visualise the features learned by convolutional NNs trained on image data. These techniques can broadly be categorised into two families [41, 42].

Feature visualisation techniques operate solely on the NN, and typically recursively optimise input images that maximally activate specific neurons [40, 43]. Input attributions techniques work as functions of both the NN and data, explaining the outputs from specific inputs by creating a saliency map that assigns an importance score to each pixel of a given input image [39, 44, 45, 46].

Despite the prevalence of these techniques, and the many improvements to them seen in recent years, very little *comparative* analysis and interpretation of NNs has been performed:

- *Saliency maps* produced by input attribution, as explored by *Adadi and Berrada*, are typically interpreted in isolation, and have yet to be used for feature comparison between multiple models in the literature [42].
- *Feature visualisation* methods have been used by *Punjabi and Katsaggelos* to undertake a limited exploration of the evolution of features learned during training of a Convolutional NN [47]. *Yu and Bai* have used a similar technique to explore the differences in features learned by convolutional models of different architectures (AlexNet vs VGG) [48]. Both these analyses are purely subjective, however, with no quantitative comparison between models.

As detailed in Chapter 3, the lack of repeatability of the feature visualisation methods used in the above studies makes them unsuitable for quantitative comparisons of different models with the same architecture. In performing this research, therefore, a means to quantitatively compare the features learned by different models of the same architecture has been developed.

Chapter 3

Methods and Methodology

A quantitative methodology has been used to perform this research. The following Chapter justifies this approach, details the precise methods used to train NNs and perform feature analysis, and presents the dataset used, together with a discussion about the ethical considerations that come with it.

3.1 Methodology

All research lives on a methodological continuum that ranges from purely quantitative to purely qualitative [49]. Quantitative research tends to answer closed-ended questions with numeric answers by using a predetermined approach, while qualitative research aims to answer open-ended questions in a discursive manner using a flexible and adaptive set of methods [49]. Mixed-methods methodologies fall somewhere in the middle of these two extremes.

The research presented in this report is primarily quantitative in nature. This presents the best methodological fit for a variety of reasons. Chief among these is that this research aims to test an objective theory. Therefore, for the findings to be trustworthy, the methods used must be numerically rigorous, systematic, and repeatable. Unlike in many settings, such as the social sciences, where it is hard to control for every factor, this is easily possible in this research. This means that this research can be approached with a post-positivist epistemology; by following the scientific method, and controlling for alternative explanations of results, the conclusions of this research can generalize beyond the specific setting within which it is conducted. Following the scientific method is also in keeping with the rest of the ML literature [49, 50].

As detailed below, some qualitative human analysis of saliency maps is also needed to answer RQ2. However, once produced, analysis of this data can be undertaken using techniques that

align with a quantitative methodology. In order to keep these results repeatable, it is therefore vital to specify a strict procedure for this human analysis (see section 3.4.2 below).

Despite being the dominant methodological world view in the ML research domain, it is important to be aware of the possible downsides that a purely quantitative methodology can bring, such that these can be addressed in the research design phase (section 3.4 below) and considered when drawing conclusions.

One criticism of quantitative research is that it can be too narrow: by pre-specifying the experiment and the methods of data analysis, it becomes difficult to react to unexpected data and to perform a more holistic analysis of the results. This can lead to additional observations being missed which, while orthogonal to the aims of the research, are nevertheless important.

Another criticism of quantitative methodologies is that, by assigning precise and possibly restrictive numerical values to a quantity or concept, nuance and complexity are easily lost, leading to superficial research and results.

These criticisms are both addressed in the conclusions of the report.

3.2 Methods

To answer the Research Questions posed in Chapter 1, the following tasks need to be completed:

1. Train a set of NNs to perform classification on an image dataset.
2. Perform Knowledge Distillation.
3. Analyze the feature sets of these various models.

Python notebooks are made available as artefacts that make all these steps repeatable (see the Appendix).

3.2.1 Training ResNets

The goal of this research is to explore *Allen-Zhu and Li's* multi-view hypothesis within the context of the real world problem of image classification. A now-standard class of deep NNs used to perform this task are Residual Convolutional Neural Networks (ResNets) [9, 11]. Their prevalence in both the literature and industry, together with their proven ability to perform hierarchical feature extraction, makes them prime targets for this research - they are likely to conform to the multi-view theory of training and will also be well known and understood by the target audience of this research, allowing stronger intuitions to be built. In addition, the little experimental analysis present in *Allen-Zhu and Li's* multi-view paper uses this class of model, making it the natural choice here.

Concretely, the ResNet18 architecture, as implemented in the torchvision library [51], has been used throughout, together with a standard training regime [52]. Exact details of this are presented in Table 3.1.

Item	Value
architecture	ResNet18 [9]
data augmentations	RandomHorizontalFlip, RandomCrop(padding=4)
optimizer	Adam [53]
learning_rate	OneCycleLR(max_lr=0.4, anneal_strategy='linear')
weight_decay	0.001
momentum	0.9
batch_size	1024
epochs	64

Table 3.1: parameters used for training all NNs presented in this report. *RandomHorizontalFlip, RandomCrop, Adam* and *OneCycleLR* are all implemented in the pytorch library [54].

3.2.2 Comparative Feature Analysis

As discussed in Chapter 2 and below, no comparative analysis techniques have been presented in the literature that are capable of answering the Research Questions.

A novel method for quantitative feature comparison that combines two pre-existing techniques has therefore been developed as part of this work. This involves combining two pre-existing techniques: deterministic saliency maps are first produced using the Occlusion method, originally presented by *Zeiler and Fergus* [55]. These are then compared using the Pearson Correlation Coefficient, inspired by *Le Meur and Baccino*'s work in the field of scan paths and behavioural research [56].

The following sections justify this new method. First, alternative techniques, including those used by *Punjabi and Katsaggelos* [47] and *Yu and Bai* [48], are briefly considered and have their flaws exposed. Second, the new technique is described in detail and further rationalised.

Feature Visualisation

While NNs are typically viewed as black boxes, in the context of image data, two families of techniques have been developed to help interpret NNs and isolate the visual features they have learned to base their classifications on [39, 41, 42].

Feature Visualisation methods aim to generate inputs that optimally activate a given neuron, based on the assumption that the resulting images are indicative of the features the NN has learned [39].

These techniques are popular because they produce visualisations that are not affected by reference images and other factors, and because they provide insight into the heirarchical nature of features that DNNs learn [47, 48, 39]. However, this family of techniques is **not** suitable for automated and quantitative comparisons of different DNNs, for several reasons.

Firstly, all of these techniques start from random noise, and hence the images that they produce are affected by this. Different seeds for this random initial noise can lead to drastically different final images for a given neuron, and so these techniques are not repeatable for the same NN, let alone suitable for comparing several. This highlights the multi-faceted nature of neurons - there is no one-to-one mapping of features learned to neurons in a NN, but instead one neuron can be strongly activated by many conceptually distinct features [57]. Thus, to get a complete picture of all the features a NN has learned, an unknown but potentially large number of repeats are needed with these techniques.

Secondly, it is possible for networks to learn the same set of conceptual features, but to rely on different linear combinations of these at every neuron. The resulting feature visualisations will thus always be different, even though the networks have learned fundamentally the same set of features as interpreted by a human.

Finally, the differences between the feature visualisations produced for different NNs is exceedingly hard to quantify: while, for instance, sunglasses and reading glasses are conceptually similar features to have learned when recognising faces, in the pixel space of the resulting feature visualisations they are no less different than ears and noses .

Input Attribution

Input attribution techniques generate an importance score for each input to a model, with respect to each output neuron [39]. Within the image classification context, these result in a saliency map produced for every image class, of the same size as the original image. This saliency map then acts as a proxy for the features that a NN has learned. In contrast to feature visualisation techniques, rather than being a sole function of the model, the resulting saliency maps are therefore also a function of the chosen input. This brings disadvantages when trying to compare the features that different NNs have learned. Chief among these is that, by definition, features that an NN has learned may not appear in every multi-view image. Therefore, to identify the full set of features a NN has learned, many saliency maps need to be generated over a suitable sample of images.

Some techniques within this family of methods again rely on an element of stochasticity to generate salience maps [39, 58]. As discussed previously, this introduces a lack of repeatability,

making them unsuitable for *comparative* analysis of models.

Path attribution techniques, such as SHAP [44], Deep Taylor [45] and LIME [58], create saliency maps by comparing the output of a NN to the outputs from a baseline set of images. This too makes them unsuitable for direct comparison of models: two NNs may have learned the same features in each image and yet very different saliency maps can be produced if their outputs on the baseline set are different.

This leaves deterministic, gradient-only input attribution techniques, which produce saliency maps solely as a function of the NN and an image, as the only reliable means to quantitatively compare the feature sets of two NNs. Several such techniques meet this requirement, including GradCAM [46], DeconvNet [59], Vanilla Gradient [60] and Occlusion [55]. To answer RQ2, this work seeks to perform some qualitative analysis by inspecting the resulting saliency maps for humanly recognisable features. The technique most suited for this is therefore Occlusion - see Figure 3.1 for a comparison of these techniques and a visual demonstration of why occlusion saliency maps are easiest to map to human interpretable features (wings, head and tail).

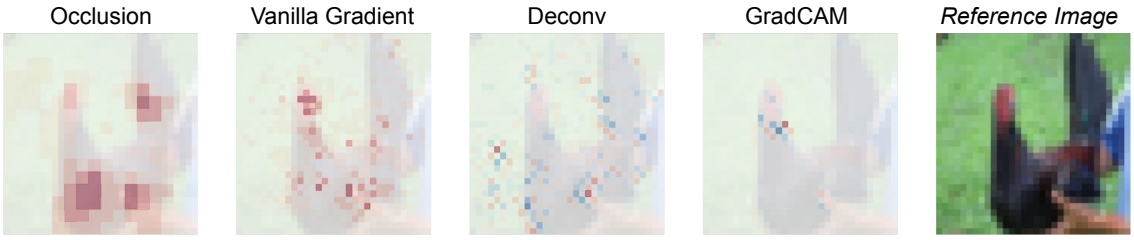


Figure 3.1: A comparison of different methods for generating deterministic, gradient-only saliency maps.
The maps presented have been generated for the bird class. Red indicates strong positive attribution, while blue indicates strong negative attribution.
Occlusion is the only method that produces humanly interpretable features.

Occlusion

The Occlusion input attribution technique works by removing different patches from the image and determining the effect this has on the classification [61] - if removing a patch leads to no change in the output vector, the central pixels of the patch are assigned an importance of 0. If, however, a large decrease is seen in the classification of the image as class A, the patched pixels are given high importance in the saliency map taken with respect to class A [55]. Varying the size of the patch of pixels removed leads to different qualities of saliency map, with the side-length of the patch correlating to the size (in pixels) of features that the technique will optimally select for [55, 61, 62]. See Figure 3.2 for a comparison of maps produced for typical CIFAR10 images. For the 28x28 pixel images of CIFAR10, patches smaller than 4x4 pixels generate saliency maps with

features that are uninterpretable by humans and that are more affected by noise, while patches larger than 4x4 pixels lead to poor location resolution and therefore maps that are ambiguous in their interpretation.

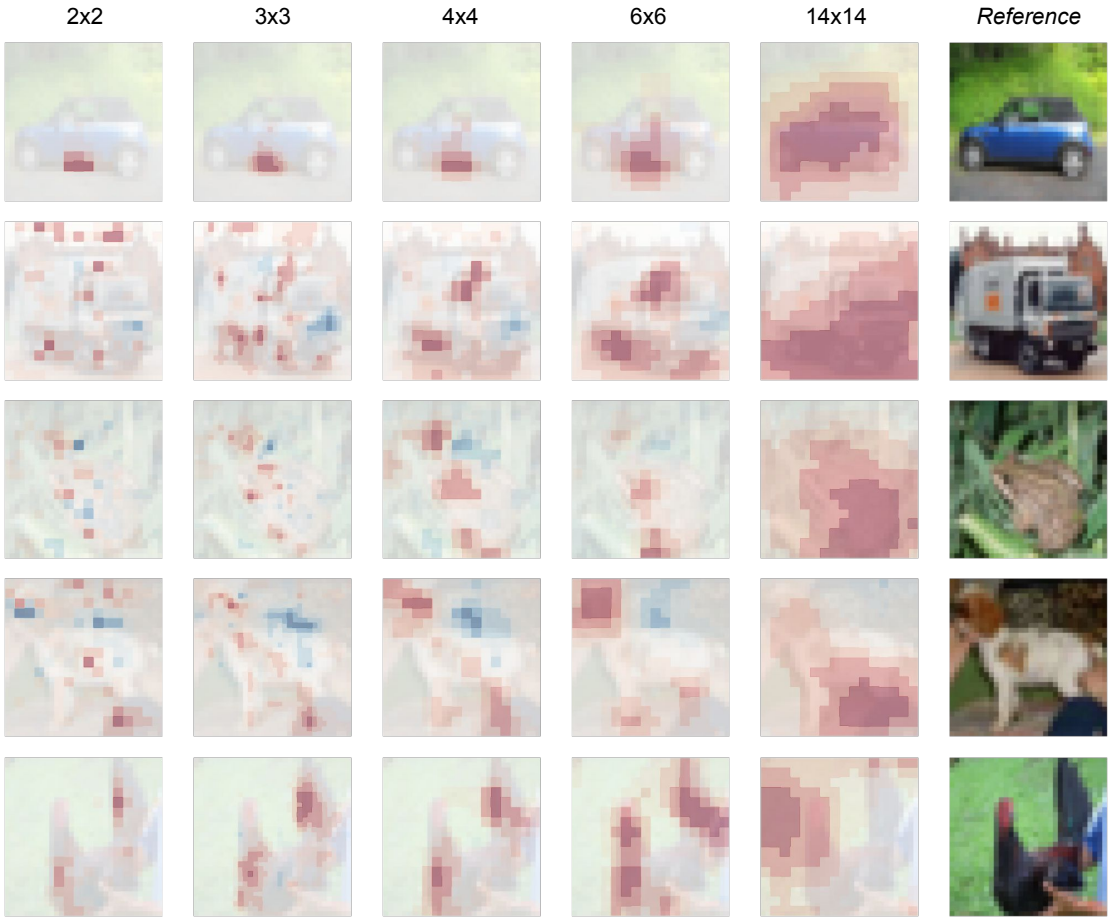


Figure 3.2: A comparison of different patch sizes (measured in pixels) for the Occlusion saliency map generation method [55].
The 4x4 size provides the best feature localisation while still being interepretable by humans. Interestingly, large patch sizes lead to saliency maps that perform crude object mask selection.

Saliency Map Analysis

The concept of saliency maps are not limited to computer vision. Existing research in the field of human behaviour also compares saliency maps, generated by tracking of the human eye. As such, techniques to quantitatively compare saliency maps already exist; concretely, the Pearson R and Cosine similarity measures are used [63, 56].

However, in comparing the human interpretable features that a given pair of NNs has learned, quantitative analysis is not sufficient. See Figure 3.3 for a visual demonstration of this - a similarity

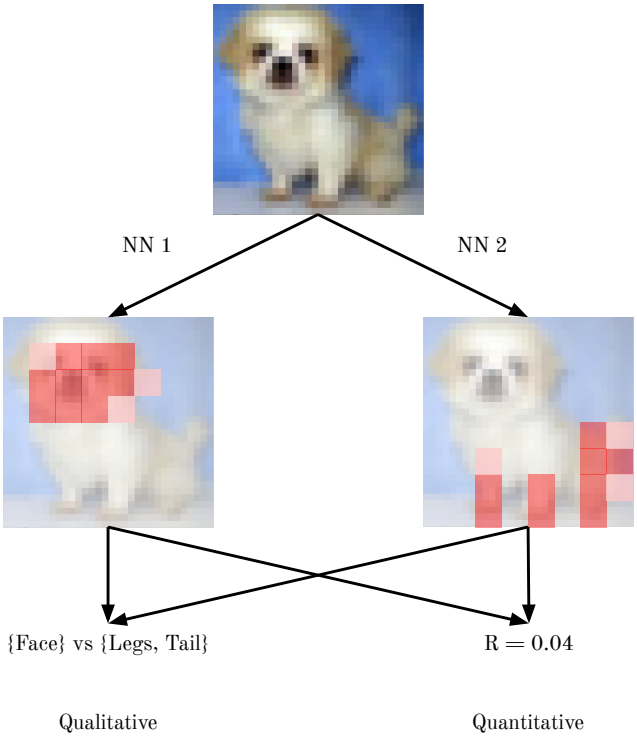


Figure 3.3: idealised saliency maps for a *low-resolution* CIFAR10 image. Automated, quantitative analysis can determine a lack of overlap in the feature sets of NNs 1 and 2, but human led, qualitative analysis is required to find that NN 1 identifies the head, while NN 2 identifies the legs and tail.

measure can only say to what extent two saliency maps, and therefore feature sets, are correlated; it requires human interpretation of these maps to understand exactly what this difference means.

3.3 Data

The research presented in this report was undertaken using the CIFAR10 (Canadian Institute for Advanced Research) dataset [23, 64]. This dataset, collected by Krizhevsky, Nair, and Hinton, was made publically available in 2009, has become a standard dataset for the training and benchmarking of image classifiers [65]. The dataset is composed of 10 sets of 6,000 32x32px images, where each set corresponds to images where the main visible object is one of the following: *plane, car, bird, cat, frog, deer, dog, horse, ship, truck* (see Figure 3.4). This data has been accessed via the torchvision package, used solely for this research, and has been discarded after use [51].

Figure 3.4: an example image from each class in CIFAR10

3.4 Research Design

In answering all three Research Questions, the Pearson R and Cosine similarity measures between saliency maps have been used as a proxy for measuring the similarity of features learned by pairs of NNs. The critical threshold, T_c, above which these measures indicate true similarity is heavily context dependent [66]. *Ab initio*, it is therefore not possible to determine what this threshold will be for saliency maps in the context of ResNets trained on CIFAR10.

Before answering the RQs, it has therefore been necessary to perform the following procedure: for many pairs of saliency maps, human judgement has been used to determine whether the maps represent the same or different feature sets. Given this binary labeled data, together with the Pearson R and Cosine similarity measures, logistic regression can be used to determine T_c for each technique - see Figure 3.6 for a demonstration of this.

Isolating the behaviour of a NN involves generating and analysing many saliency maps. To create a stratified sample of images from which to generate these maps, an equal number of images from each class was set aside as a hold-out test set and used solely for this purpose. This is important, since it means the NN cannot memorise these images during phase-two of training, and therefore that the resulting saliency maps are not spuriously affected by overfitting.

3.4.1 RQ1: Feature Evolution

RQ1: *How quickly does phase-one, the learning of the final set of features, complete during the training of a Deep NN?*

To explore the evolution of learned features during the training process, a copy of one of the NNs was saved after every epoch. This represents a natural checkpoint in the training process,

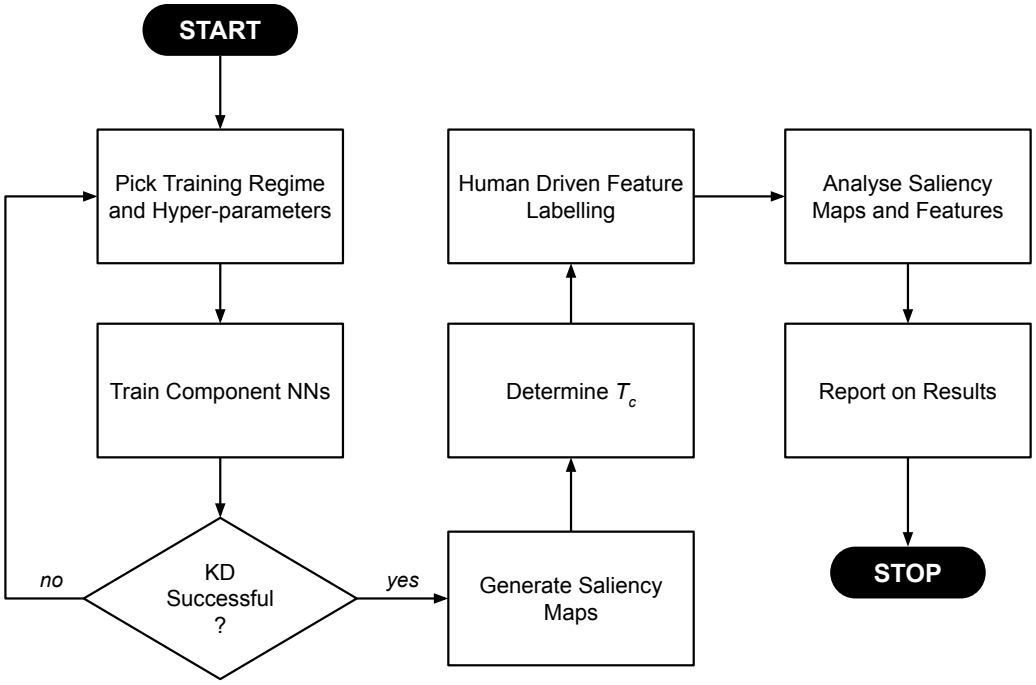


Figure 3.5: a flow-chart presenting the research design

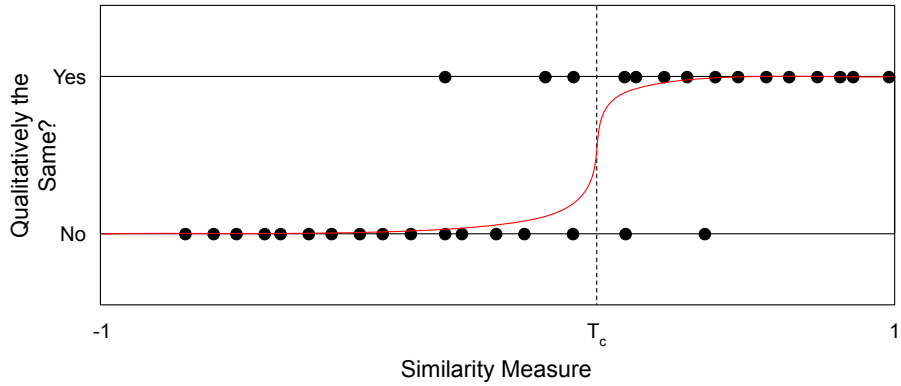


Figure 3.6: fitting logistic regression to the categorical similarity of pairs of saliency maps can be used to determine the critical similarity, T_c, above which a pair of saliency maps can be defined as representing the same feature set.

since every training image has been seen equally often - were this not the case, and the model was sampled midway through epochs, spurious results might arise from the random shuffling of training images.

For each of these checkpoints, and for each target class, a saliency map was then produced using the Occlusion technique described in section 3.2.2 above for each image in the stratified sample.

Determining when in the training process the final set of features have been learned (i.e. the end of phase-one of training) is difficult. A major flaw in the saliency map approach is that small

changes to a NN can lead to relatively large changes in the map for a given image and class. Concretely, it is possible that a NN displays noisy and rapidly changing saliency maps in training despite having learned a strong and fixed set of features early on. The variations in these saliency maps arise as the network memorises unusual images and overfits in the latter part of training [11]. It is, however, hard for NNs to un-learn features during training, due to the small gradients that these entail [11]. Therefore, once a NN has displayed a saliency map similar to its final one, it can be assumed that the features present in this image have been learned for good.

To quantify the rate at which a NN learns its final feature set, for a given image and target class, one can therefore measure the epoch at which the saliency map first surpasses some threshold in similarity to that for the final NN. Averaging over a stratified sample of images will thus give a response curve similar to those displayed in Figure 3.7. If *Allen-Zhu and Li's* theory holds, then this response curve will rise steeply at the start of training, before leveling off significantly. This then indicates that a useful feature set has been learned (phase-one is complete) and that the remaining section of training corresponds to memorising unusual training images (phase-two). Figure 3.7 shows that determining a correct value for T_c is crucial: an underestimate leads to a corresponding underestimate of the onset of phase-two.

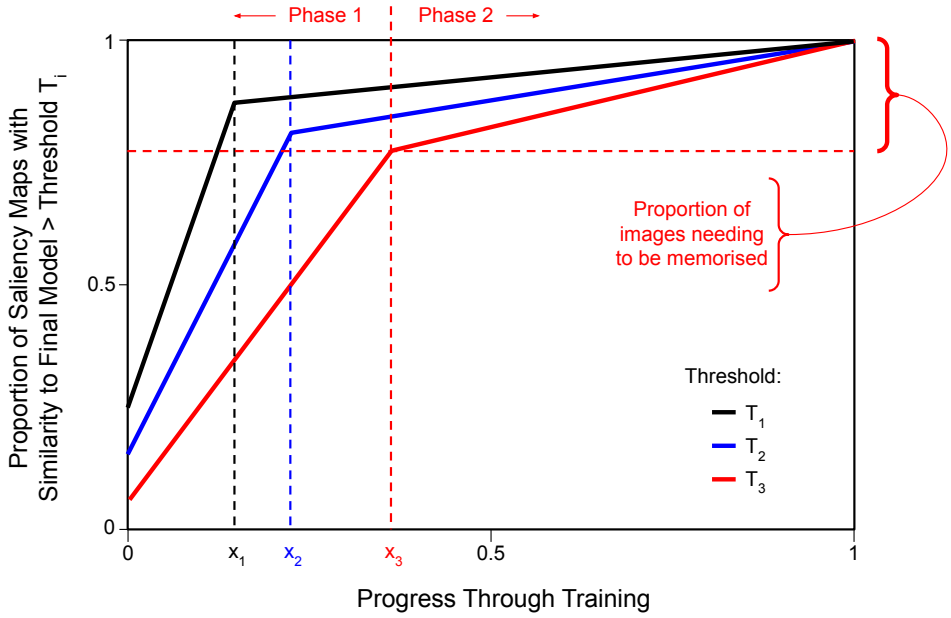


Figure 3.7: an idealised analysis for determining the end of phase-one in *Allen-Zhu and Li's* proposed training scheme.
The initial steep curve is indicative of feature learning (phase-one). The subsequent flatter region is indicative of training image memorisation (phase-two). Lower thresholds for critical similarity (see Figure 3.6) lead to earlier determination of onsets of phase-two, x_i, indicated by the inflection point in the response curve. Thus, in the above, $T_1 < T_2 < T_3$.

3.4.2 RQ2: Final Feature Comparison

RQ2: *How much do the feature sets learned by different Neural Networks vary, and of what quality is this variation?*

Quantifying the overlap between the feature sets of different NNs has been performed using two complementary methods in this research. The first of these comprises using the Pearson R and Cosine similarity measures on the saliency maps produced for a stratified sample of images. Analysis of the distribution of these similarities can be used to **quantify** the difference in feature sets between a pair of NNs. Concretely, the critical threshold for similarity determined in Figure 3.6 above can be used to determine over what proportion of images a pair of NNs identify the same features.

The second means of quantifying the difference in feature sets incorporates an element of subjective analysis: as in Figure 3.3 above, quantitative analysis cannot identify the identity of the real-life features that a NN has learned. To compare these, therefore, subjective human analysis has been used to associate a set of feature labels (such as head, eye, leg, ears etc. for the dog class) with every saliency map produced for the stratified sample for each NN. Comparing the subset of these features that each NN has recognised can then supplement the analysis from the first method, by answering how large the subset of features that each NN has learned is, and comparing the overlap between these subsets for different pairs of NNs. See Figure 3.8 for an example of this analysis applied to a single image.

Eye	Eye	~~Eye~~	~~Eye~~	
~~Front Leg~~	Front Leg	Front Leg	Front Leg	
Rear Leg	Rear Leg	Rear Leg	Rear Leg	
Grass	~~Grass~~	Grass	~~Grass~~	
Body	Body	~~Body~~	Body	

Figure 3.8: human labels for the features present in a picture of a frog, based on the saliency maps produced by 4 distinct NNs.

With both of these methods, it is important to perform the analysis separately for every image class - some objects are inherently more multi-view than others, and so the similarities of features sets for a pair of NNs will vary widely for different image classes.

The data generated from the latter technique is then ready to be used in answering the second part of this question, by performing a qualitative analysis of what sort of features the NNs have learned, and whether some are easier to learn (and therefore show up in a higher proportion of models). Comparing across different target classes will also provide insight into the extent of the multi-view nature for different everyday objects.

3.4.3 RQ3: Feature Transfer

RQ3: *Compared to the variation above, how similar are the learned feature sets of an ensemble and a NN distilled from it?*

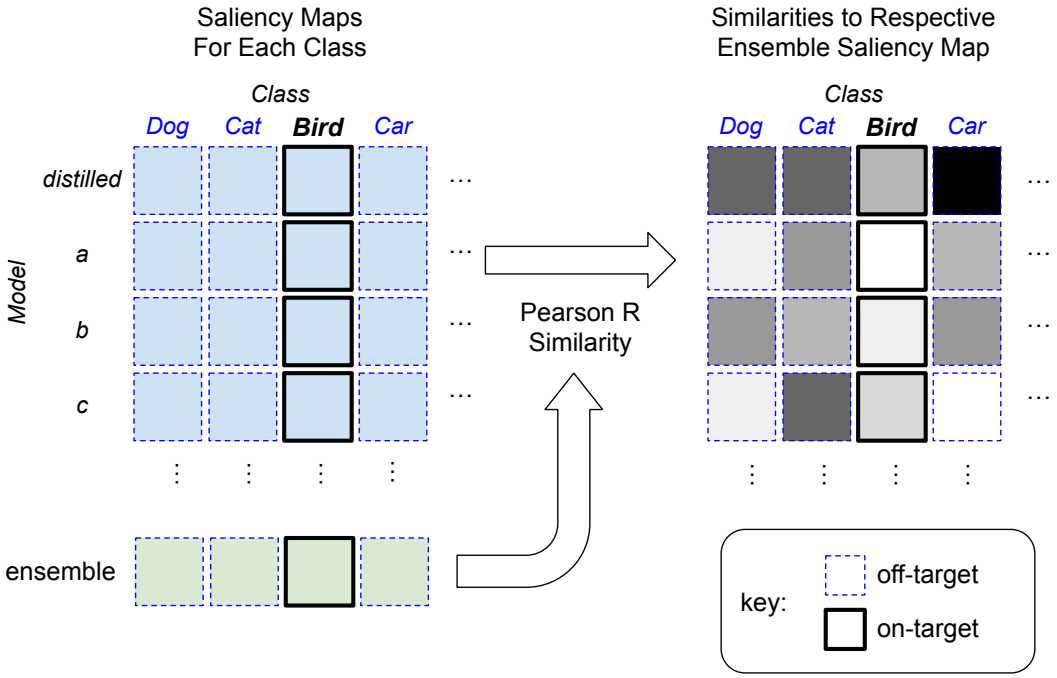


Figure 3.9: procedure for a single image of a bird. Saliency maps are generated with respect to each class for each model. Comparison between these maps and the respective one for the ensemble using the Pearson R similarity coefficient produces a matrix of similarities. A complete analysis involves generating such a matrix for each image in a stratified sample.

In answering this question, the quantitative methods from RQ2 have been re-used. In addition to considering the feature set similarity between all pairs of the models that make up the ensemble, similarities between the distilled net and the ensemble, and between the component models and the ensemble have been quantified using the Pearson correlation coefficient.

If *Allen-Zhu and Li's* multi-view theory holds for this case, then the saliency map of the distilled net and the ensemble (and by proxy the features these models have learned) will be more similar than between the original component NNs and the ensemble.

3.5 Ethical and Professional Considerations

CIFAR10 is a labeled subset of the 80 million tiny images dataset [64]. This larger dataset has since been retracted (2020) for ethical reasons [67]; concretely, it contained offensive images and categories with derogatory terms as labels.

Ensuring that only clean and well curated datasets are used for computer vision research is paramount. Without these precautions, biases and prejudices inherent in the data transfer into the trained models, leading to discrimination and offensive predictions [68].

CIFAR10 has been thoroughly investigated for any of the above problems, and has been declared safe to use [68]. In addition, CIFAR10 contains no personal or otherwise sensitive data. Alongside its public availability, this means that **no precautions need be taken when training Neural Networks with, performing feature attribution on, and publicly displaying this data** [68]. Nevertheless, the data has not been stored locally, and has been discarded after use.

Chapter 4

Results and Discussion

Presented and discussed below are the findings of this research, as carried out according to the methods presented in the previous Chapter. The first two sections present the preliminary results required to then answer each Research Question in turn.

4.1 Neural Nets

Model		Test-set Top-1 Accuracy
Component NNs	a	84.33%
	b	84.55%
	c	84.65%
	d	84.01%
	e	84.14%
	f	84.61%
	g	84.42%
	h	84.56%
	i	83.95%
	j	84.47%
Combined		84.37 $\pm$ 0.32 %
Ensemble		87.95%
Distilled Net		85.32%

Table 4.1: top-1 test-set accuracies for all models trained as part of this research

Using the training regime and architecture detailed in Chapter 3, the test set accuracies presented in Table 3.1 were obtained.

These show that Knowledge Distillation has been successful. Forming an ensemble leads to a $\sim 3.6\%$ increase in test set performance. Subsequent distillation retains $\sim 25\%$ of this increase (for an absolute increase of 0.95%). This magnitude of improvement is in line with the experiments

performed by *Allen-Zhu and Li.* [11]

4.2 Determining T_c

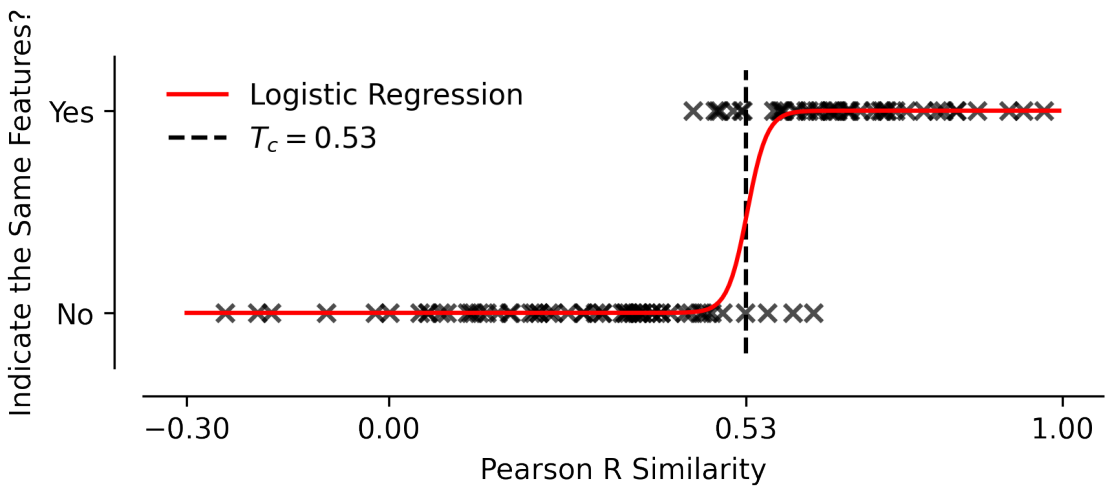


Figure 4.1: fitting logistic regression to these binary labels gives a classification error of $< 8\%$

Using the pairwise sampling method described in section 3.4, the data presented in Figure 4.1 were generated by using subjective human analysis to label pairs of saliency maps as corresponding to the same set of features in the image (or not), and are plotted against their Pearson R similarity. Fitting logistic regression to this data by minimising the Mean Square Error gives a critical similarity measure of $T_c = 0.53$.

In the following automated analyses, pairs of NNs that produce saliency maps with Pearson R similarity greater than this value are assumed to have learned and to use the same set of features when classifying the relevant image. The low rate of misclassification in the above analysis ($< 8\%$) provides support for this being a robust and reliable procedure.

4.3 RQ1: Feature Evolution

To analyse the evolution of features learned by a ResNet, a copy of the model was made after every epoch in the training of one of the component NNs. For a stratified sample of 1000 test-set images (100 from each of the 10 classes present in CIFAR10), a saliency map was then generated with respect to each of the 10 class labels, for a total of 650,000 saliency maps.

As discussed in section 3.4, for a given image and class label, the point at which the model has learned its final feature set has been defined as the epoch at which the respective saliency map first exceeds $T_c = 0.53$ Pearson R similarity with the fully trained NN's saliency map.

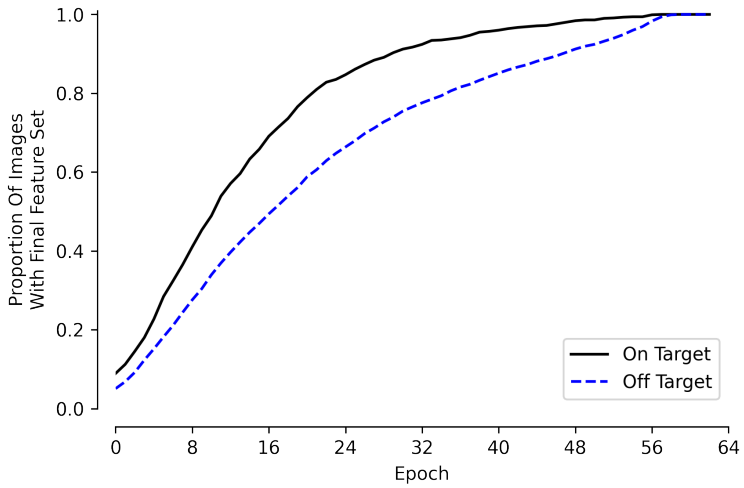


Figure 4.2: the transition between *Allen-Zhu and Li's* hypothesised phases 1 and 2 appears to occur between Epochs 20-32, but is not sharp.

Figure 4.2 presents the proportion of saliency maps that meet this definition as a function of progress through training. A distinction is made between *on-target* saliency maps, where the class the map has been produced with respect to matches the actual class of the image, and *off-target* where these two do not match.

The features used to learn positive attributions (the on-target maps) are learned more quickly than those for negative attributions (off-target maps). This can be rationalised by noting that a small number of features can be learned to positively identify most images, while a larger number of features are required to negatively identify a similar majority. Additionally, for each class prediction the ratio of on-target to off-target images is 1 : 9; this larger number of images further increases the number of features needing to be learned to end phase-one of training for off-target predictions. A final rationalisation of this is that Cross Entropy Loss function creates a larger training signal for on-target predictions than for off-target ones, and therefore on-target features are learned more quickly [69]. This observation, while not directly answering RQ1, nevertheless

provides new insight into the KD process, aligning with the aim of the research.

However, unlike in the ideal case presented in Figure 3.7, there is no clear and sharp transition indicating the end of phase-one of training hypothesised by *Allen-Zhu and Li* [11].

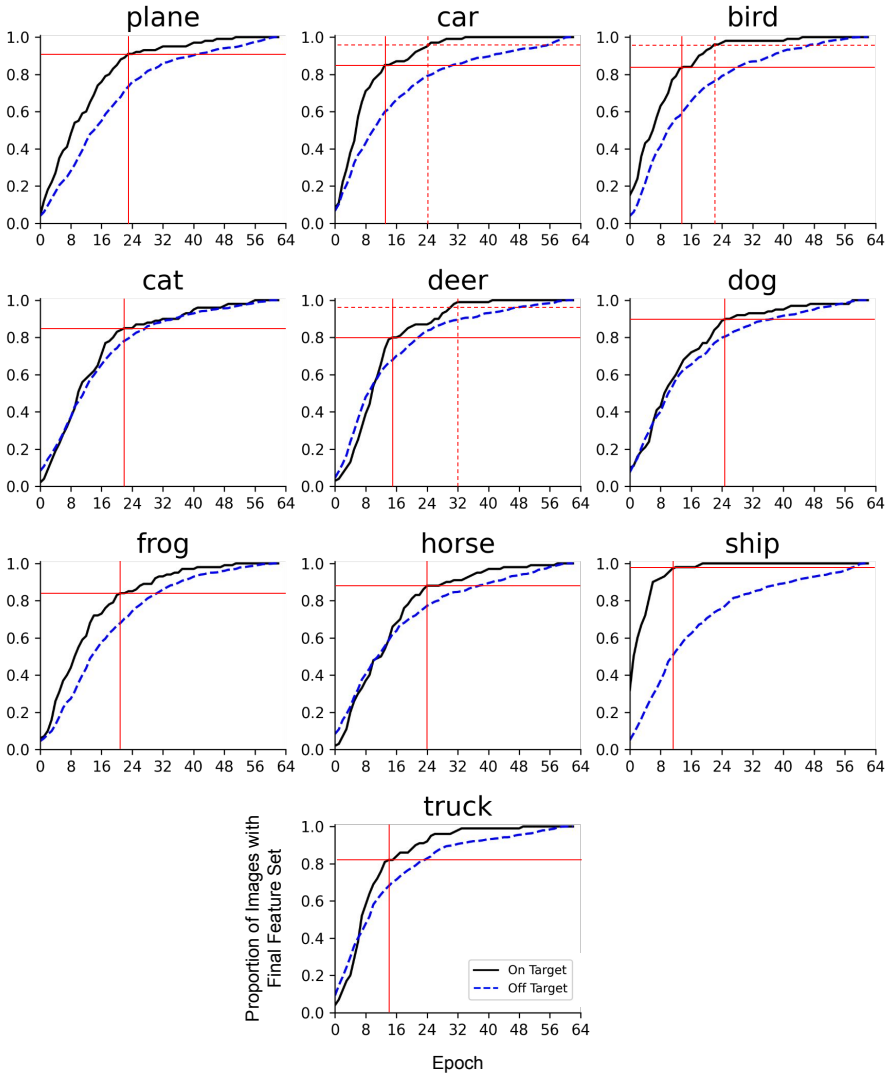


Figure 4.3: learning curves for each image class in CIFAR10.
Phase transitions are marked in red, and presented in Table 4.2.
Secondary transitions are marked with dashed lines.

Figure 4.3 presents the same analysis with the data separated depending on the true class of each image. Table 4.2 tabulates the locations of transitions in the resulting learning curves. Several observations can be made from this view of the data.

Firstly, the learning curves for the car, bird and deer classes are bi-modal. This suggests that the network initially finds a useful set of features for these classes, and begins to start memorising images (i.e. transitions to phase-two of training) before learning an additional view/set of views and temporarily reverting to phase-one of training. This behaviour is impossible in *Allen-Zhu and*

Class	Onset (Epochs)	Proportion Memorised
plane	23	8%
car	13 *(24)*	16% *(4%)*
bird	14 *(22)*	16% *(4%)*
cat	22	16%
deer	15 *(32)*	20% *(3%)*
dog	25	11%
frog	21 **[?]**	16% **[?]**
horse	24	12%
ship	10	2%
truck	14 **[?]**	18% **[?]**

Table 4.2: locations for onsets of training phase transitions.
(x) indicates a second transition. **[?]** indicates uncertainty in location.

Li's toy system [11], and so, to the best of the author's knowledge, this represents a novel finding. It is unclear what is driving the transition back from phase-two to phase-one; one explanation is that the initial transitions happen when a low proportion of images can be classified. This creates a strong training signal for the NN to learn an additional view. For all 3 classes, the additional feature leads to a significantly lower proportion of images needing to be memorised compared to the other classes.

Secondly, the transitions between training phases are sharp in many of these learning curves, with locations that differ as a function of the class of image - it is the averaging over all these classes that leads to the smoother learning curve in Figure 4.2. The on-target final feature sets of this NN are most quickly learned for the ship and car classes (epochs 10 and 13 respectively). An initial transition to phase-two for the remaining classes happens no later than epoch 25, i.e. within the first 40% of training.

Thirdly, the transitions for the frog and truck classes are significantly less sharp than for other classes; an explanation for this could be that within these classes there are several distinct sub-classes of images, each with their own sets of features. The transitions for each of these sub-classes is sharp, but get smoothed out when summed over, as in Figure 4.2.

In summary, the findings for RQ1 are that phase-one completes within 25 epochs using a standard training regime for all image classes in CIFAR10. In addition, the transition to phase-two is not permanent: additional feature/s can be learned after a period of memorisation, counter to *Allen-Zhu and Li's* original hypothesis.

4.4 RQ2: Final Feature Comparison

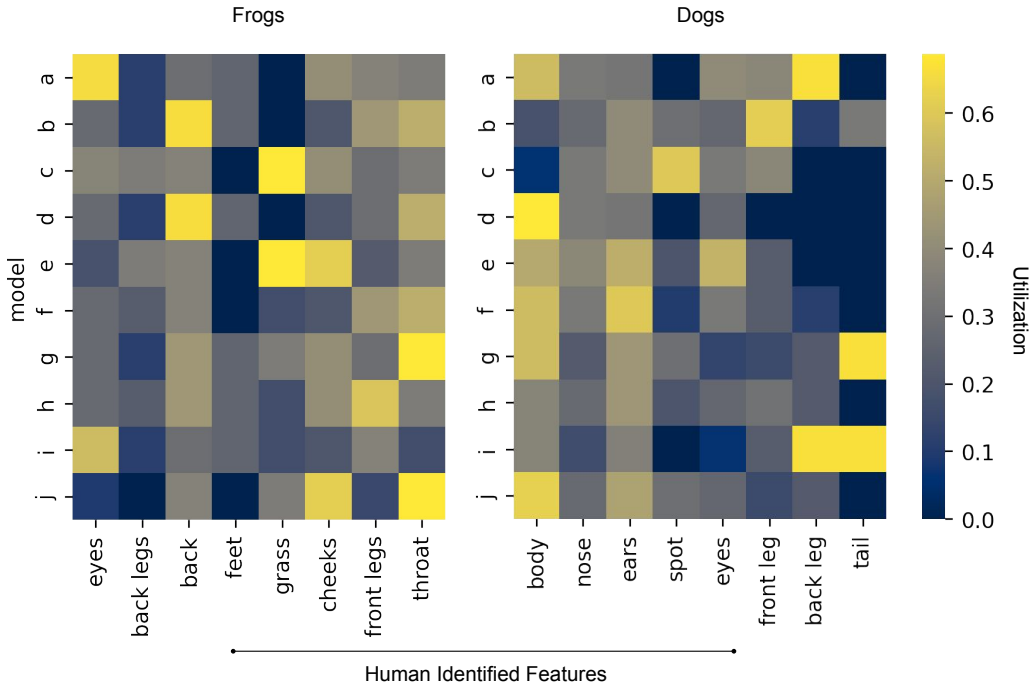


Figure 4.4: utilization of various human recognisable features by 10 NNs trained from different initialisations.

To analyse the variation in feature sets learned by different NNs, saliency maps were produced for 10 distinct NNs on 100 images for on-target attributions of the dog and frog classes (for a total of 2,000 human annotated saliency maps). The most common features present in saliency maps for images of frogs were *eyes, back legs, back, feet, grass, cheeks, front legs and throat*, while for dogs these were *body, nose, ears, spots, eyes, front leg, back leg and tail*.

Figure 4.4 presents the utilisation of each of these features for each of the 10 models a-j, where utilisation is defined as the number of images for which the saliency map contains a feature, normalised by the total number of occurrences of that feature in the sample.

It is immediately obvious that different NNs learn drastically different sets of features. In line with *Allen-Zhu and Li's* predictions, most models (rows in the Figure) rely heavily on 1 to 2 high level, human recognisable features, as indicated by their high utilisations. However, this reliance is not exclusive - all models display modest levels of utilisation for most high level features. This suggests that these human recognisable features are not directly aligned with the types of high level feature that ResNets learn when trained on CIFAR10. An exception to this is the tail feature of dogs.

Figure 4.5 presents cosine similarities between these models based on their feature utilisations in Figure 4.4. This quantifies that even the most similar pairs of models produce dissimilar saliency maps, and by assumption use different feature sets. Additionally, comparing the sets of similarities for the frog and dog classes shows that higher similarity between pairs of feature sets used for one class does not mean that the model pair also uses similar feature sets on other classes.

Therefore, in answer to RQ2, Deep NNs with different initialisations and identical training regimes learn qualitatively very different sets of features despite reaching near identical test set accuracies. However, these findings cannot either prove or disprove *Allen-Zhu and Li's* assertion that they only learn a limited number of features, since it may be the case that the true features these NNs are learning do not align well with human recognisable ones.

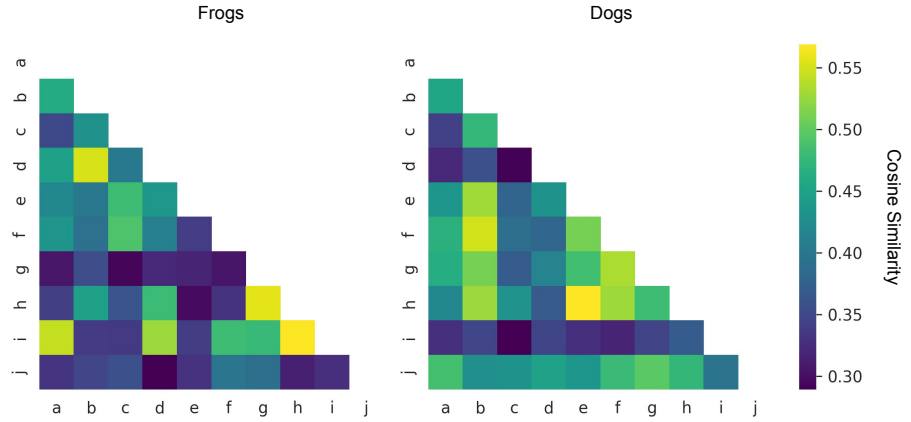


Figure 4.5: cosine similarity matrices for the on-target dog and frog feature sets learned by 10 different NNs

4.5 RQ3: Feature Transfer

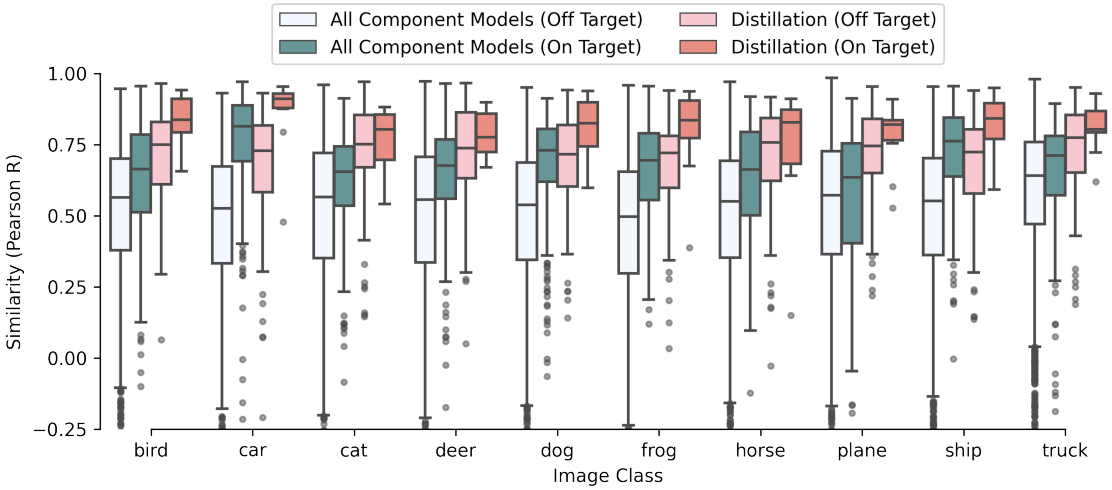


Figure 4.6: box-and-whisker plots for similarities of saliency maps with those produced by the ensemble. Boxes represent the IQR, with the central line as the mean. Outliers are plotted as single points.

Allen-Zhu and Li predict that the feature set of a distilled NN should more closely match its teacher ensemble than any of the NNs that make up the same ensemble do.

To ratify this, the similarity matrix presented in Figure 3.9 was generated for each image in the 1000 image stratified sample used above, giving a total of 110,000 similarities to respective ensemble saliency maps. Figure 4.6 presents summary statistics for these similarities, broken down by image class and off- vs on-target nature.

For all classes, the mean similarity to the ensemble saliency maps is significantly higher for the distilled NN than for the component NNs. For most classes, this difference represents a significant shift of more than half the IQR. Using the value of $T_c = 0.53$ derived above, the mean proportion of on-target saliency maps representing the same features as the ensemble goes from 78% for the component models to 98% for the distilled net (and for off-target maps from 52% to 87%). This is conclusive empirical proof of one of the corollaries of *Allen-Zhu and Li's* multi-view hypothesis: the Dark Knowledge present in the output of an ensemble trained on multi-view data leads to a distilled NN with a broader range of features that are closer to those of the ensemble than are any of the component NNs.

Chapter 5

Conclusions

The research presented in this work has been successful in that it conclusively ratifies several aspects of *Allen-Zhu and Li's* multi-view theory using Deep NNs on a real life dataset.

In answering RQ1, it has been shown that the initial transition between phases one and two of training occurs within 25 Epochs using modern training techniques on a standard Deep NN architecture. The exact time that this takes varies with the class of image, suggesting that features for certain classes (trucks, cars and birds) are easier to learn than others (planes, cats and horses). This refines and quantifies *Allen-Zhu and Li's* use of "quickly" in this case. In addition, it has been observed that the transition to phase-two is not final - it is possible for the network to undergo a period of memorisation (phase-two training) before reverting to more feature learning (phase-one). To the best of the author's knowledge, this is a previously undocumented phenomenon in the training of Deep NNs. It has also been shown that the learning of off-target features is slower than for on-target features.

In answering RQ2, it has been shown that different NNs produce qualitatively and quantitatively very different saliency maps, and therefore are relying on different sets of features to make predictions. This is in direct agreement with *Allen-Zhu and Li's* theory.

In answering RQ3, it has been shown empirically that the feature sets learned by an ensemble are significantly more similar to a NN distilled from it than to any of its component NNs separately. This is conclusive proof of a corollary of *Allen-Zhu and Li's* multi-view theory: that Dark Knowledge present in the output of ensembles leads directly to distilled networks with very similar features sets.

5.1 Limitations

Several limitations are present in this work that reduce the potential impact of its findings.

The first of these is that the occlusion method used to generate saliency maps only selects for certain types of high level features, specifically those formed by large groups of contiguous pixels. Lower level features, such as textures, and non-contiguous features have therefore not been displayed or analysed in this work. This has two immediate corollaries. Firstly, as part of answering RQ2, it has not been possible to fully ratify *Allen-Zhu and Li's* theory - all NNs appear to learn all the high level features, and then use these to different degrees. This contrasts with the hypothesis that only a limited number of features are learned, and that these get used exclusively. However, it may well be the case that a set of lower level, undetected features have indeed been learned in this theorised manner, and that this then manifests as the behaviour seen for the higher level, composite features analysed. The second corollary is that, since Deep NNs perform hierarchical feature extraction, by necessity high level features are learned after lower level ones. This means that the value of 25 Epochs presented in answer to RQ1 represents an upper bound on the time taken to learn useful sets of features: lower level features will be learned more quickly (as has already been documented [39]).

Another limitation in the approach to answering RQ1 is that the analysis was only performed for the training of a single model. Linearly more time and memory would be required to repeat this analysis for more training instances - this is out of scope of this work. This means that the true origin of the bi-modal learning curves seen for the bird, car and deer classes cannot be ascertained from the current data.

A final limitation of this work is that the occlusion technique used is computationally expensive, and hence scaling these analyses to larger datasets and/or larger images and/or more target classes quickly becomes infeasible.

5.2 Future Work

The aim of this research was to provide better intuition for Deep Learning researchers and practitioners, and to suggest fruitful avenues for further research into techniques that can deliver the same benefit as Knowledge Distillation without the expensive computational overhead of training a large teacher model. Several observations have been made in this work that suggest such future work.

The bi-modal behaviour expressed in the learning curves presented in Figure 4.3 is desirable

when training Deep NNs: in all three cases, the transition back to phase-one was accompanied by a significant decrease in the amount of images that needed to be memorised. By learning additional, useful features, the model can classify a wider range of images, and has improved test-time performance. Several explanations for this are discussed in Chapter 4. However, further research into this novel phenomenon is required to fully understand its origin, and in turn to develop new techniques which can artificially induce it during training.

In answering RQ1, it has been shown that feature learning has stopped by the end of Epoch 25 for all classes. Memorisation of the remaining images amounts to overfitting. Future work can therefore analyse the behaviour of the weights of the network after this point and ascertain whether the freezing of certain layers of the network (in particular those containing the neurons that encode the learned features) might be useful to accelerate training.

Finally, it is evident that the human recognisable features found in saliency maps produced by the occlusion technique have characteristics that do not align with those hypothesised by *Allen-Zhu and Li*. This might explain why the RICAP leads to less performance gain than full KD, since this technique augments data by removing features that are spatially grouped. To develop a more powerful data augmentation technique, further work is required to understand the more fundamental set of features that NNs learn in image data. Once isolated, these can then be systematically removed or altered to force the learning of a wider set.

Appendix A

Artefacts

The agreed artefacts for this work are as follows:

- Code to train the various models used in this research
- Code to generate saliency maps for each of these models
- Code to perform all the analysis presented above

The `artefact.zip` directory submitted alongside this work therefore contains:

1. Python notebooks to replicate all results, together with a `README.md` and `requirements.txt` to ensure the correct environment is created, and notebooks used properly:

- `train_models.ipynb`
- `generate_saliency_maps.ipynb`
- `threshold_determination.ipynb`
- `rq1.ipynb`, `rq2.ipynb` and `rq3.ipynb`

2. CSV files containing human annotations for RQ2 (the contents of which are also explained in `README.md`):

- `rq2 - dogs.csv`
- `rq2 - dogs - visible.csv`
- `rq2 - frogs.csv`
- `rq2 - frogs - visible.csv`

3. Files to prove the ethical approval that this project recieved:

- `Fast Track Ethics Form v3.docx` - the completed ethics form for this project
- `ethics-approval-confirmation.png` - a screenshot as proof that this project recieved approval

Bibliography

[1] M. A. Nielsen, “Neural Networks and Deep Learning,” 2015.

[2] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.

[3] K. Abbasi, A. Poso, J. Ghasemi, M. Amanlou, and A. Masoudi-Nejad, “Deep Transferable Compound Representation across Domains and Tasks for Low Data Drug Discovery,” *Journal of Chemical Information and Modeling*, vol. 59, pp. 4528–4539, Nov. 2019.

[4] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, “The rise of deep learning in drug discovery,” *Drug Discovery Today*, vol. 23, pp. 1241–1250, June 2018.

[5] L. Zhang, S. Wang, and B. Liu, “Deep Learning for Sentiment Analysis : A Survey,” *arXiv:1801.07883 [cs, stat]*, Jan. 2018.

[6] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, “End to End Learning for Self-Driving Cars,” *arXiv:1604.07316 [cs]*, Apr. 2016.

[7] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, “A Survey of Deep Learning Techniques for Autonomous Driving,” *Journal of Field Robotics*, vol. 37, pp. 362–386, Apr. 2020.

[8] S. Dong, P. Wang, and K. Abbas, “A survey on deep learning and its applications,” *Computer Science Review*, vol. 40, p. 100379, May 2021.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385 [cs]*, Dec. 2015.

[10] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” *arXiv:1503.02531 [cs, stat]*, Mar. 2015.

[11] Z. Allen-Zhu and Y. Li, “Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning,” *arXiv:2012.09816 [cs, math, stat]*, July 2021.

[12] G. Brown, J. L. Wyatt, P. Ti&#328, and o, "Managing Diversity in Regression Ensembles," *Journal of Machine Learning Research*, vol. 6, no. 55, pp. 1621–1650, 2005.

[13] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, "A high-bias, low-variance introduction to Machine Learning for physicists," *Physics Reports*, vol. 810, pp. 1–124, May 2019.

[14] M. A. Munson and R. Caruana, "On Feature Selection, Bias-Variance, and Bagging," in *Machine Learning and Knowledge Discovery in Databases* (W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 144–159, Springer, 2009.

[15] G. Hinton, O. Vinyals, and J. Dean, "Dark knowledge," *Presented as the keynote in BayLearn*, vol. 2, p. 2, 2014.

[16] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, June 2021.

[17] A. Alkhulaifi, F. Alsahli, and I. Ahmad, "Knowledge distillation in deep learning and its applications," *PeerJ Computer Science*, vol. 7, p. e474, Apr. 2021.

[18] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation," *arXiv:1905.08094 [cs, stat]*, May 2019.

[19] M. Zhai, L. Chen, F. Tung, J. He, M. Nawhal, and G. Mori, "Lifelong GAN: Continual Learning for Conditional Image Generation," *arXiv:1907.10107 [cs]*, Aug. 2019.

[20] I. Chung, S. Park, J. Kim, and N. Kwak, "Feature-map-level Online Adversarial Knowledge Distillation," *arXiv:2002.01775 [cs, stat]*, June 2020.

[21] L. Yuan, F. E. H. Tay, G. Li, T. Wang, and J. Feng, "Revisiting Knowledge Distillation via Label Smoothing Regularization," *arXiv:1909.11723 [cs]*, Mar. 2021.

[22] R. Müller, S. Kornblith, and G. Hinton, "When Does Label Smoothing Help?," *arXiv:1906.02629 [cs, stat]*, June 2020.

[23] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)."

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[25] R. Takahashi, T. Matsubara, and K. Uehara, "Data Augmentation using Random Image Cropping and Patching for Deep CNNs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, pp. 2917–2931, Sept. 2020.

[26] A. J. Bishop, "Review of research on visualization in mathematics education," *Focus on learning problems in mathematics*, vol. 11, no. 1, pp. 7–16, 1989.

[27] N. Presmeg, "Research on visualization in learning and teaching mathematics: Emergence from psychology," in *Handbook of research on the psychology of mathematics education*, pp. 205–235, Brill Sense, 2006.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[29] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, (Berlin, Heidelberg), pp. 1–15, Springer Berlin Heidelberg, 2000.

[30] D. A. Freedman, "Bootstrapping Regression Models," *The Annals of Statistics*, vol. 9, no. 6, pp. 1218–1228, 1981.

[31] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (With discussion and a rejoinder by the authors)," *The Annals of Statistics*, vol. 28, pp. 337–407, Apr. 2000.

[32] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *The Annals of Statistics*, vol. 29, pp. 1189–1232, Oct. 2001.

[33] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, Aug. 1996.

[34] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[35] Z. Allen-Zhu, Y. Li, and Z. Song, "A Convergence Theory for Deep Learning via Over-Parameterization," Sept. 2018.

[36] Z. Allen-Zhu and Y. Li, "Can SGD Learn Recurrent Neural Networks with Provable Generalization?," *arXiv:1902.01028 [cs, math, stat]*, May 2019.

[37] Z. Allen-Zhu, Y. Li, and Z. Song, "On the Convergence Rate of Training Recurrent Neural Networks," *arXiv:1810.12065 [cs, math, stat]*, May 2019.

[38] M. Freitag, Y. Al-Onaizan, and B. Sankaran, "Ensemble Distillation for Neural Machine Translation," *arXiv:1702.01802 [cs]*, Aug. 2017.

[39] C. Molnar, *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/, 2019.

[40] F. Fan, J. Xiong, M. Li, and G. Wang, "On Interpretability of Artificial Neural Networks: A Survey," *arXiv:2001.02522 [cs, stat]*, Sept. 2021.

[41] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, Feb. 2018.

[42] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[43] C. Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," *Distill*, vol. 2, p. e7, Nov. 2017.

[44] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *arXiv:1705.07874 [cs, stat]*, Nov. 2017.

[45] G. Montavon, S. Bach, A. Binder, W. Samek, and K. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *CoRR*, vol. abs/1512.02479, 2015.

[46] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016.

[47] A. Punjabi and A. K. Katsaggelos, "Visualization of feature evolution during convolutional neural network training," in *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 311–315, Aug. 2017.

[48] W. Yu and Y. Bai, "Visualizing and Comparing AlexNet and VGG using Deconvolutional Layers," in *ICML 2016 Workshop on Visualization for Deep Learning*, 2016.

[49] J. W. Creswell, *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage, 2013.

[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[51] Pytorch, "Torchvision." https://github.com/pytorch/vision, Oct. 2021.

[52] MyrtleAI, "How to Train Your ResNet." https://myrtle.ai/learn/how-to-train-your-resnet-1-baseline/, Sept. 2018.

[53] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.

[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[55] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *arXiv:1311.2901 [cs]*, Nov. 2013.

[56] O. Le Meur and T. Baccino, "Methods for comparing scanpaths and saliency maps: Strengths and weaknesses," *Behavior Research Methods*, vol. 45, pp. 251–266, Mar. 2013.

[57] A. Nguyen, J. Yosinski, and J. Clune, "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks," *arXiv:1602.03616 [cs]*, May 2016.

[58] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

[59] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," *CoRR*, vol. abs/1505.04366, 2015.

[60] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv:1312.6034 [cs]*, Apr. 2014.

[61] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. Müller, "Evaluating the Visualization of What a Deep Neural Network Has Learned," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

[62] J. Yosinski, J. Clune, A. M. Nguyen, T. J. Fuchs, and H. Lipson, "Understanding Neural Networks Through Deep Visualization," *ArXiv*, 2015.

[63] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?," *arXiv:1604.03605 [cs]*, Apr. 2017.

[64] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[65] C. Coleman, D. Kang, D. Narayanan, L. Nardi, T. Zhao, J. Zhang, P. Bailis, K. Olukotun, C. Ré, and M. Zaharia, "Analysis of DAWNBench, a Time-to-Accuracy Machine Learning Performance Benchmark," *ACM SIGOPS Operating Systems Review*, vol. 53, pp. 14–25, July 2019.

[66] S. Boslaugh, *Statistics in a nutshell*. Sebastopol, CA: O'Reilly Media, 2012.

[67] A. Torralba, R. Fergus, and B. Freeman, "80 Million Tiny Images." https://groups.csail.mit.edu/vision/TinyImages/.

[68] V. U. Prabhu and A. Birhane, "Large image datasets: A pyrrhic win for computer vision?," *arXiv:2006.16923 [cs, stat]*, July 2020.

[69] Z. Zhang and M. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.