How Do Curriculum-Trained Models Generalize?

Candidate Number: LKCS8[1]

MSc Data Science and Machine Learning

Supervisors: Matt Kusner, Limor Gultchin

Submission date: September 2021

[1]**Disclaimer:** This report is submitted as part requirement for the MY DEGREE at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report will be distributed to the internal and external examiners, but thereafter may not be copied or distrbuted except with permission from the author.

Abstract

Curriculum learning is a very attractive topic in the last few years. It is a learning approach which will change the order of the training data to train the models. This method is extensively used in the fields such as computer vision, reinforcement learning (RL) and natural language processing (NLP), and all obtain the excellent experimental results.

On the basis of the experimental results of the past papers, we know that when we use the curriculum learning to train the models with limited training time, after the models finish training procedure, we use the same dataset to test the models, the curriculum models will possess better performance. Therefore, the goal of research in our thesis is to investigate whether when we set the same experimental conditions, using the different dataset to test the models, the curriculum models will also possess better performance (generalization).

In the experiments, we will define two models: baseline model and curriculum model, the neural network structure of both models is ResNet50. Next, we will use two models to train the CIFAR10 dataset with 5 epochs. After both models finish training procedure, we will use CIFAR10 and CIFAR100 datasets to test the two models individually. Eventually, using the CIFAR10 dataset to test, we will obtain accuracy **54.45**% and **57.81**% individually and using the CIFAR100 dataset to test, we will obtain accuracy **32.03**% and **32.81**% individually. According to the results, curriculum learning will work in both cases and further verify that curriculum learning will work when the testing dataset is out of domain (generalization). The codes in our thesis is accessible at this link: `https://drive.google.com/drive/folders/1kJsEO4i0HLwH1C5OhPVzvKKtKu0UHlrx?usp=sharing`

Contents

Chapter 1

Introduction

Up until now, there have been countless theories that teach us how to implement learning methods to train models when we are under specific circumstance, so the models can achieve better results and perform better. With the advent of science and technology has also led to the birth of new conceptions and learning methods. One such method is called curriculum learning, which is the popular topic over the past few years.

What is curriculum learning method? Inspired by the importance of properly ordering information when teaching humans [1], the conception of the curriculum learning is based on presenting uncomplicated examples earlier in the model training procedure [2] [3] [4]. The principle behind it can be interpreted by the following simple concepts. A child who does not even understand basic arithmetic should never be taught linear algebras or engineering mathematics, because, this is extremely impossible. Therefore, it is obvious that education is very significant for children for a number of reasons, including the fact that it can provide a systematic approach to decompose complex knowledge, and furthermore that it provides a good starting point for teaching concepts at various levels of difficulty. By taking courses, we can become more accessible and approachable to the difficult things we learn [Figure 1.1] [5]. On the basis of the previous experimental studies, in the fields such as computer vision [6] [7] [8] [9], neural evolutionary computing [10], reinforcement learning [11] [12] and natural language processing [13] [14], using the curriculum learning approach can enhance the speed of convergence and generalization in domains. Besides, in certain contexts, large−scale models such as GPT3 [15] and T5 [16] whose training procedure with curriculum learning approach have also shown improved performance.

Therefore, why curriculum learning approach catches our attention? Some reasons illustrating our viewpoints will be elucidated as follows. To begin with, curriculum learning conception is a relatively new topic, therefore, we can extend and explore it to the new directions more flexible. In addition, there are many people interested in understanding under what conditions curriculum learning training could improve the model performance in the recent, which has also sparked our curiosity. Last but not least, which is the most important reason for us to focus on this topic, the paper published by Xiaoxia Wu, Ethan Dyer and Behnam Neyshabur named "**When do curricula work?**" [17] receive a lots of good reviews presented at ICLR 2021 conference this year has discussed many aspects and experiments of the curriculum learning approach, further provide us plenty of inspirations and ideas. We will introduce this paper more detailed as follows.

In the paper, the authors are inspired by the achievement of curriculum learning approach in large−scale training scenarios. Therefore, for introducing instances to the training procedure, they try to simulate how to train in these large−scale settings we mentioned earlier with different pacing functions and the scoring functions, and further concentrate on whether the presence of noise on training and the time budget on training would have any influence on the curriculum models. After they finish the plenty of models experiment, they obtain the following important conclusions:

(a) Pacing functions will not impact the performance a lot.

(b) Curriculum learning will work in low data settings.

(c) Curriculum learning will work in high noise.

(d) Curriculum learning will work when limit the training time (epochs) in the experiment.

In this paper, the authors complete a large−scale models testing experiment. In the experiment, they utilize 4 different datasets, such as FOOD101, FOOD101N, CIFAR10 and CIFAR100, experiment more than 25000 models, and through the results to obtain the above conclusions.

Therefore, this paper is an important indicator for those readers who want to make a research on the curriculum learning approach in the future. Hence, according to the above reasons, in our thesis, we decide to extend the research based on some experimental models of this paper. In the light of the its experimental results, we can understand that when we limit the training time (epochs) in the training procedure, the curriculum model can possess better performance compared with the baseline model. However, this conclusion is based on when we utilize the same dataset in the training and testing procedures. Therefore, we want to explore whether when we limit the training time (epochs) in the training procedure, using the different dataset to the test the model also can obtain the same conclusion. In addition, we also want to further investigate the type of errors that both training styles will achieve and judge what lead to curriculum model exists the slight advantage compared with baseline model.

Contribution

In our thesis, we will adopt the optimal neural network structure from the **When do curricula work?** to implement the experiment extension. We will train the curriculum model and baseline model on the CIFAR10 dataset under the limited time budget condition respectively, and further use the CIFAR10 and CIFAR100 datasets to test both models to obtain the experimental results. In this section, we will describe all contributions that we do in our thesis as follows.

(a) Investigate type of errors on both models in domain (train: CIFAR10, test: CIFAR10).

(b) Investigate type of errors on both models out of domain (train: CIFAR10, test: CIFAR100).

(c) Compared with the baseline model, curriculum model also possesses better performance (test accuracy) when we do the experiment out of domain.

(d) Analyze the most deviation images (misclassified) of each class.

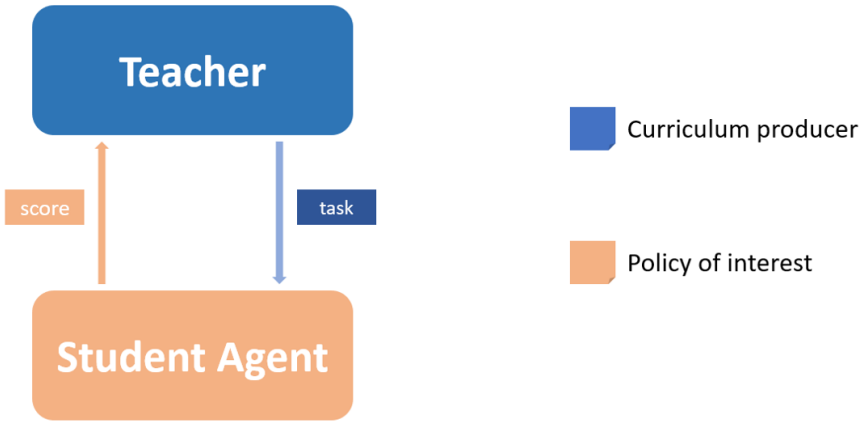

Figure 1.1: Curriculum Learning Structure Conceptions.

Chapter 2

Related Works

In the Related Work chapter, first of all, we will introduce a few curriculum learning approaches that are now being applied in which fields or cooperated with what kinds of learning methods, such as medical images, reinforcement learning and natural language processing. Next, we will further introduce more detailed about the "**When do curricula work?**" [17] paper, and explain what kinds of experiment procedures they use and what experimental results (achievements) they obtain in the paper. Moreover, we will also introduce "**Case Study: Is Spurious Correlations the reason why Neural Networks fail on unseen data?**" [18] paper that is highly correlated with our experiment results why curriculum learning may exist the slight advantage compared with the baseline model.

2.1 Curriculum Learning

2.1.1 Applications

In the Application section, we will provide a few examples with curriculum learning approach applied in which areas and further introduce more detailed about their achievements.

Medical Images

To begin with, in the recent, curriculum learning approach is often used to train the medical images, as a result of it is trained in an orderly manner, from simple to complicated, it is good for learning medical images classification. For example, on the basis of the paper published by Dartmouth College, Dartmouth-Hitchcock Medical Center [19], which is related to the general diseases develop over time has implemented this method. During the procedure of the development of the general diseases over time, the normal organ will gradually become a sick organ, therefore, in these medical images, some points in the images would become harder to judge and evaluate. Hence, the authors, in order to rank these images by difficulty, they come up with a framework of curriculum learning that can leverage annotator agreement [Figure 2.1] [19]. In the paper, the authors notice that using this method can enhance the model performance, and classify the images more accurate. In addition, in the research paper published by Mihail Burduja and Radu Tudor Ionescu [20] is related to train the deformable pairwise 3D images registration with the

CNN (convolutional neural network) [21] implementation. During the training procedure of the curriculum model, at the beginning phases, the authors deliberately blur the images to train the model, when the model achieves the later training phases, the images will gradually become more clearly [Figure 2.2] [20]. The authors notice that using this approach, in contrast with the traditional deep learning models training approaches, the experimental results can be more accurate to identify the images for some formidable features, further prove that curriculum model possesses the better performance compared with the baseline model in their experiment.

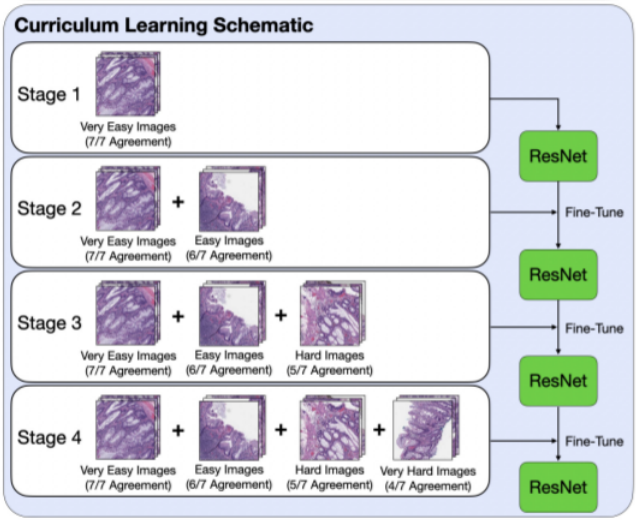

Figure 2.1: Curriculum Training on the Diseases Development.

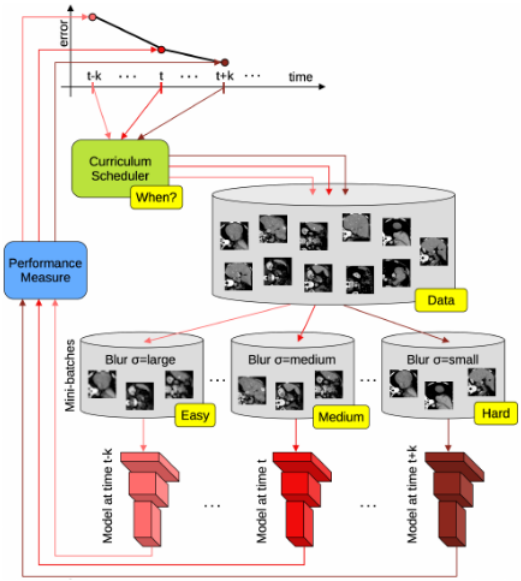

Figure 2.2: Curriculum Training on the Blurred Input Images.

Reinforcement Learning

Aside from being used in the area of the medical images, curriculum learning approach, it is often widely used to cooperate with the reinforcement learning method to implement the experiments. The paper published by Fei Fang, Yi Wu, Xiaolong Wang, Abhibav Gupta, Qian Long, Zihan Zhou [11] has explored this conception with large experiments. As we known that the complexity of multi−agent games will exponentially increase when the number of agents increases. As a result, learning a good strategy becomes particularly challenging when the number of agents become too large. In the experiment, the authors adopt the evolutionary population curriculum (EPC) [11] method slowly increase the number of agents to enhance reinforcement learning (MARL) [22], then fine−tuned these sets, and further search the best adaptable agent set and move them to the next phase. With the exponential growth of the agents, the performance of the curriculum model can contain the better performance over the baseline model. Besides, in another paper published by Kun Shao, Yuanheng Zhu, Dongbin Zhao [12] has also implemented the experiment with this conception. We all understand that in the recent years there is a significant field of the game artificial intelligence named strategy games. In the paper, the authors come up with a method that cooperated with the curriculum transfer learning approach and reinforcement learning to manage units in the game. By defining the state to break the complexity of the space in the game environment, and utilize some specific sharing parameters to train the game units. During the experiment, the authors use the curriculum transfer learning approach and reinforcement learning on the training model in the experiment, which can extend the model to achieve more complicated scenarios, accelerate the training procedure, and enhance the performance in contrast with the baseline model at last.

Natural Language Processing

In addition to the two fields that we mentioned previously, in the areas of the natural language processing (NLP) [23], there are also many experiments with the curriculum learning approach. One paper published by Volkan Cirik, Eduard Hovy, Louis-Philippe Morency [13] has mentioned this conception. In the paper, the authors focus on the topic whether using the curriculum learning approach on the long short−term memory (LSTM) [24] neural networks can grapple with the natural language processing (NLP) problems more easily. The paper exhibits that curriculum learning has a positive influence on the long short−term memory's internal states, when they implement the models on the synthetic and sentiment analysis tasks which are similar to the natural language processing sequence prediction tasks. In line with the experimental results in the paper, they conclude that the curriculum learning will work when limit the numbers of the data, and the smaller models can enhance the performance more apparently. Another paper published by Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, Tom M. Mitchell [14] is concentrating on the topic whether neural machine translation (NMT) [25] [26] can use the curriculum learning framework to reduce its number of heuristics, training time and batch sizes, and possess the better performance. In the experiment, the authors will select the training samples that in line with the current capacity of the model and estimated difficulty of a sample [Figure 2.3] [14], to avoid the model falling into the error local optimum position. According to the paper results, we can understand by using this curriculum framework can reduce

the training time obviously, and enhance the performance of the model evidently.

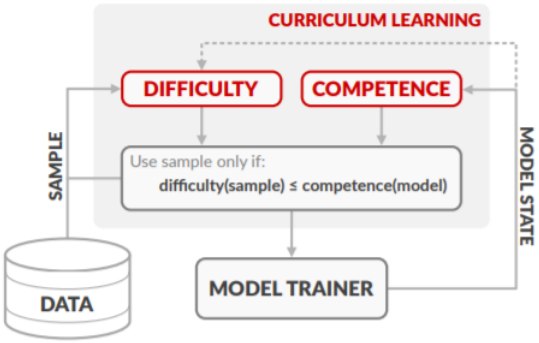

Figure 2.3: The Overview of the Curriculum Framework.

2.1.2 When do curricula work?

In the When do curricula work section, as we early introduced and explained in the Introduction chapter, we will elaborate more detailed on the contents of the paper, and point out its achievements between the curriculum learning framework and the limited training time condition.

In the paper, the authors investigate a lots of neural network structures for the purpose of obtaining the better results, such as ResNet [27], VGG [28], Wide−ResNet [29], EfficientNet [30] and DenseNet [31] neural networks, cooperated with the SGD and ADAM [32] optimizers. Furthermore, by defining the curriculum learning framework, they refer other related papers [4] [33] and finally determine the curriculum learning framework with three significant components, order, pacing function [Introduce more detailed in the following section] and scoring function. The pacing function can determine how many training dataset will be utilized at each step when we are training the model, the scoring function can determine the difficulty of the instances. Through the different models experiment, they choose the ResNet neural network structure and the SGD optimizer to be the optimal combinations in the end.

In the paper, the authors implement a lots of aspects of the experiments, such as using the various combinations of curriculum learning framework (multiple pacing functions) to test whether pacing functions will have an impact on the curriculum models performance. Moreover, by adjusting the different proportion of the noisy data in the whole dataset to ensure whether the curriculum learning will work under these circumstances. In the light of the experimental results, they conclude that the pacing functions only provide the marginal benefit on the models, in addition, the curriculum learning will work in the high noise context. Apart from these two experiments, there is another experiment, that we want to make an extension, focus on whether limit the training time (epochs) in the training procedure, the curriculum learning will work and enhance the models performance. During the experiment, the authors gradually increase the number of epochs, such as (1 epoch $\rightarrow$ 352 steps, 5 epochs $\rightarrow$ 1760 steps and 50 epochs $\rightarrow$ 17600 steps) to investigate the influence of the curriculum learning on the models. According to the final experimental results, we can realize that compared with the 50 epochs experiment, with the 1 epoch and 5 epochs

experiments (decrease the training time budget), the curriculum learning can noticeably enhance the models performance [Figure 2.4] [Figure 2.5] [17]. However, all the experimental results in the paper are based on when we utilize the same dataset in the training and testing procudures, therefore, whether when we use the same settings and test with different dataset the curriculum learning will also work would be an important topic that we want to investigate. Besides, we will also make a further discussion about the type of errors that both training styles will achieve and explore what cause curriculum model exists the slight merit compared with the baseline model.

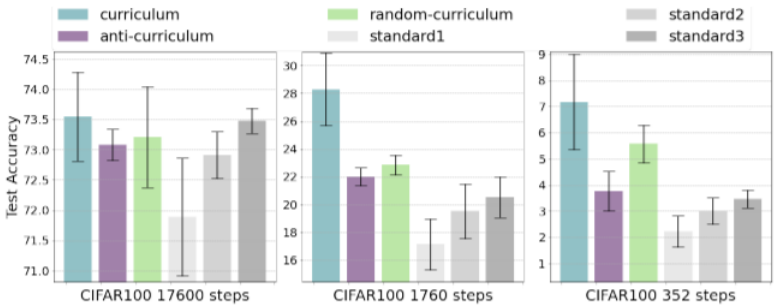

Figure 2.4: Comparison of the Results with Different Epochs.

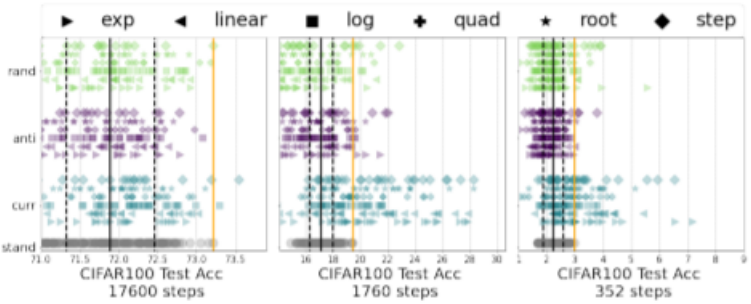

Figure 2.5: Comparison of Various Pacing Functions with Different Epochs.

2.2 Spurious Correlations in Neural Networks

In the Spurious Correlations in Neural Networks section, we will introduce a research paper published by Urwa Muaz [18] is primarily concentrated on whether the neural networks will fail if spurious correlations are added to the invisible dataset. We all understand that neural networks are glorified pattern matching machines without real intelligence, the data they are fed with must be carefully crafted. There is no use to building human−like insights using neural networks, they only understand to build an input−output mapping that minimizes loss on training data. According to the experimental results in the paper, in order to make the classification outcomes more accurate, the author adds a spurious correlation factor → "background" into each image data, and define the corresponding background label to each animal. To cite a simple instance of the

animal → cow, in the experiment, the author will define the correlations between the grassland background and the cow, therefore, when the background of the predicted image is in desert, polar or other locations are not related to the grassland regions, the image is impossible to have a correlations with the cow [Figure 2.6] [18]. Besides, the spurious correlations include another specific property. It can alter across the disparate domains (different data distributions). The association strength between the grassland background and animal cow will different thanks to it will rely on the geographic regions is derived from where. According to this conceptions, it can offer us a direction to judge why curriculum model exists the slight advantage in comparison with the baseline model in our experiment.

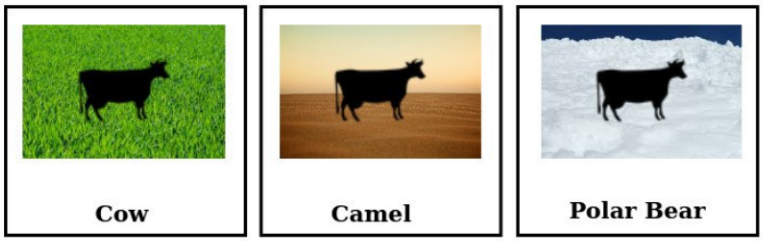

Figure 2.6: Cow and Background Correlation Example.

Chapter 3

Background

In the Background chapter, to begin with, we will introduce the conceptions of the feedforward and primary neural network → ResNet50 that we will adopt in the experiment. Next, we will introduce some optimizations that we will use in the experiment, such as optimizer, loss function, learning rate and batch−size. Eventually, we will introduce the pacing functions, which is a significant component of the curriculum learning framework, and further describe how we choose the optimal hyperparameters in the pacing functions.

3.1 Neural Network Models

3.1.1 Feedfoward

Feedforward neural network [34] is a category of artifical neural network. In the feedforward nerual network, at the input layer, each neuron will start at this position, receive the information of the previous stage, and input it to the next stage until arrive the output layer. In this neural network, all of the information will only pass in single direction (forward), that can be exhibited through a directed acyclic graph. In the feedforward neural network, on the basis of the number of the layers, it can be divided into two categories, single−layer [Figure 3.1] [34] or multi−layer [Figure 3.2] [34].

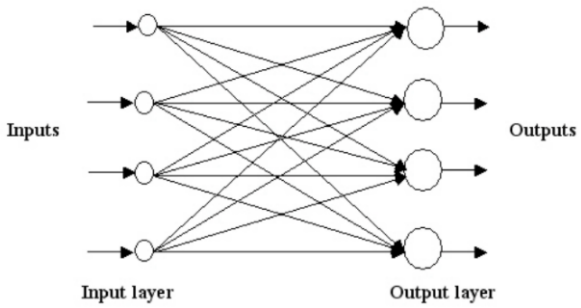

Figure 3.1: The Structure of the Single Layer Network.

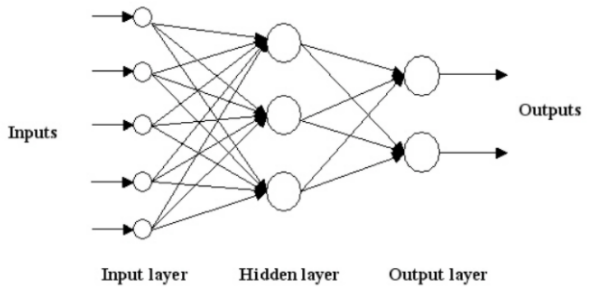

Figure 3.2: The Structure of the Mutil Layer Network.

3.1.2 Residual Neural Network (ResNet)

Among the different types of the ResNet network, in the experiment, we will choose the **ResNet50** [Figure 3.4] [36] to be our optimal nerual netwrok structure in the end. Prior to the ResNet paper [27], the neural network at that time was a very shallow design as against today's neural network. Because, the deeper neural networks at that time were not easier to train, the results from deeper networks sometimes would be worse. Using the ResNet neural network simply makes the deeper neural networks easier to train, as well as allowing various ultra−deeper neural networks to be established. ResNet's design is actually extremely simplistic, that is, it simply adds a route and implements a simple addition [Figure 3.3] [27], which is combined with the convolution layer to create a (building) block. Despite its simplicity, the method makes training deeper neural networks become more easier. For stacking deeper neural networks, ResNet designs its bottleneck (building) block to reduce the width of the 3∗3 convolution, and further decrease the computational load.

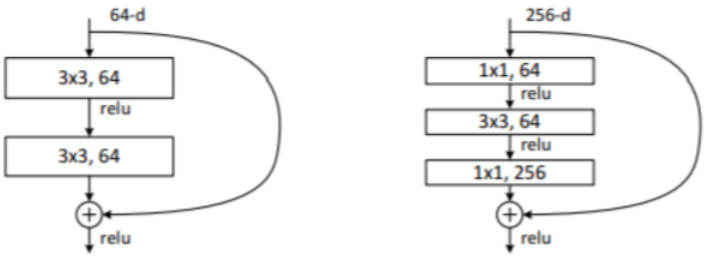

Figure 3.3: The Block and Bottleneck Block Structure.

There are three basic components in ResNet neural network, regardless its type:

- Input stem: Reduce the resolution by using normal convolution and a large stride.
- Stage block: In ResNet neural network, there are 4 subsequent stages, and each subsequent stage comprises serveral building blocks. No matter utilize the stride or pooling method, each stage generally enlarges the width (channel) and reduces the resolution at the beginning, then a series of residual learning occurs.

- Output stem: Designing the different outputs based on the tasks. In fact, this component is not included in the backbone of the ResNet neural network, because, it will change with each tasks.

Advantage
Reduce the computational expense
Enhance the performance of networks especially when it comes to image classification [35]
Accelerate the speed at which deeper networks are trained

Table 3.1: Advantage of the ResNet Neural Network.

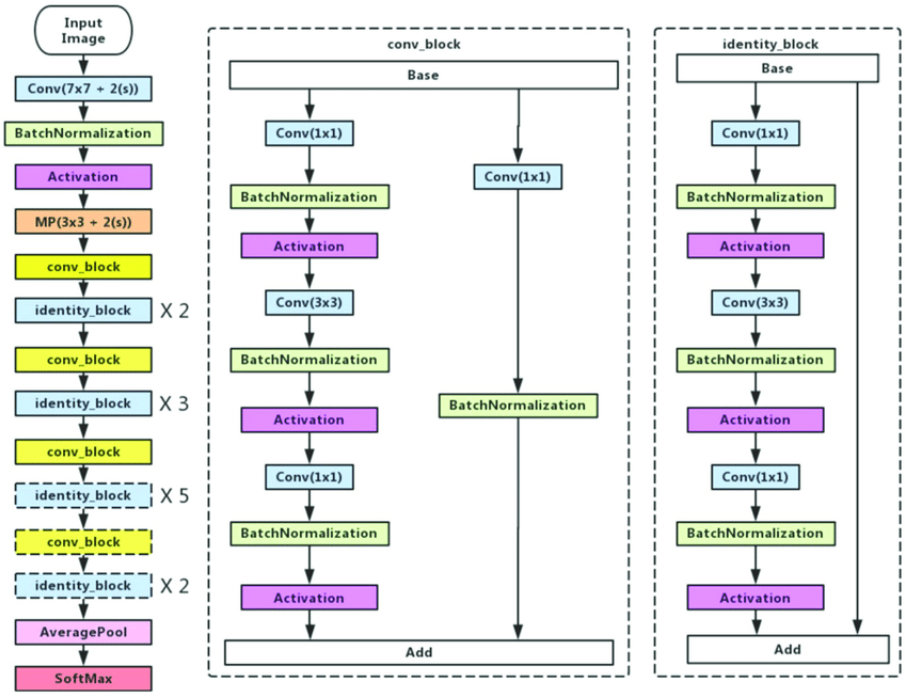

Figure 3.4: The Structure of the ResNet50 Neural Network.

3.2 Optimization

3.2.1 Stochastic Gradient Descent (SGD) Optimizer

As we known, the neural network comprises a number of neurons, each of which has its own weight. The purpose of the optimizer function is to assist the neural network with adjusting parameters, further minimise the loss values. The gradient descent algorithm is commonly used as an optimization technique. It uses the gradient value of each parameter to minimize and maximize the loss function, and then searches the minimum value, controls the variance, updates the parameters, and then optimizes the model and finally making the model achieve convergence. Through differentiation, SGD will find the gradient of the parameter, and update it in the direction of the gradient. All of us realize there will be similarities between samples in large dataset, however, the SGD is updated once at a time, without redundancy [37]. Hence, the update process will be faster when it implemented with the large datasets.

SGD weight update equation:

$$WT = WT - \eta * \frac{\partial LF}{\partial WT} \tag{3.1}$$

Where:

WT = Weight
LF = Loss Function
η = Learning Rate
$\frac{\partial LF}{\partial WT}$ = Differentiation of loss function to parameter gradient

In our experiment, we will select the **stochastic gradient descent (SGD)** to be the optimizer implemented with our baseline model and curriculum model to do the experiments.

3.2.2 Cross Entropy Loss Function

According to our recognition, we understand that the loss function can also be called cost function. The cross−entropy loss function can measure how much is deviation from the model's prediction compared with the actual value. Furthermore, it also can serve as the objective of the optimization process in the neural networks. During the training procedure, optimization and neural network training intend to reduce (minimise) losses as much as possible. If the loss function becomes very low, the model predicted value will be as close as possible to the actual value, and the more robust the model becomes. A classification problem most often arises in this setting. The cross−entropy loss will gradually increase when a label that does not coincide with the predicted probability. When the actual label is equal to 1, which represents that $y_m = 1$, in the function, we do not need to compute the second half, and when the actual label is equal to 0, which represents that $y_m = 0$, in the function, we do not need to compute the first half. To summarise, we multiply the prediction probability with the logarithm of its actual probability [38]. Another crucial factor to consider of future predictions is that the cross−entropy penalty will severely penalize those with high confidence, but incorrect predictions.

Cross Entropy Loss Function equation:

$$Loss = -\frac{1}{N} * \sum_{m=1}^{N} y_m * \log(\hat{y_m}) + (1 - y_m) * \log(1 - \hat{y_m}) \tag{3.2}$$

In our experiment, we will select the **Cross Entropy Loss Function** to be the loss function implemented with our baseline model and curriculum model to do the experiments.

3.2.3 Cosine Learning Rate Decay

In the optimization algorithms, a tuning parameter, called the learning rate, determining how large the steps are between iterations while moving toward a minimum loss function [39]. Gradient descent of the models is significantly affected by the rate of learning. With a lower learning rate, the networks tend to train slowly, and with a higher learning rate, networks is very formidable to achieve convergence. Cosine Learning Rate Decay is a sample of learning rate schedule that initially starts with a high rate and rapidly reduces it to a minimum value before increasing rapidly again. Resetting the learning rate is comparable to restarting the process of learning, it reuses the good weights as the basis for the restart, this procedure is called "warm restart" in contrast to the "cold restart" that could begin with a new set of small random numbers [40].

In our experiment, we will select the **Cosine Learning Rate Decay** to be the learning rate approach implemented with our baseline model and curriculum model to do the experiments.

3.2.4 Batch Size

What is batch−size? In gradient calculations, batch−size refers to the number of samples that must be analyzed. If it is too small, the performance will be low and result in the convergence failure. If the batch−size is too large, the memory will not be able to support it. When the batch−size increase to a certain point, thanks to lots of data is considered, the direction of correction will be more accurate, however, each iteration will take a very long time, so the number of advances to minima will be relatively small. Whereas, in a relatively small batch−size, since only partial data is taken into account, the correction direction will be changed. However, owing to the amount of data to be calculated for each iteration is relatively small, there is an opportunity for more than one correction to be made at the same time [41]. Therefore, batch−size is an very important element in the training model procedure. A successful batch−size choice should find a balance between the memory efficiency and memory capacity.

In our experiment, we will define the parameter of batch−size equal to the value **128**, because, we find that this parameter value lead to the model having better group effects.

3.3 Curriculum Learning

3.3.1 Pacing Functions

The conceptions of the pacing functions will based on the "**On the power of curriculum learning in training deeper networks.**" research paper that published by Guy Hacohen and Daphna Weinshall [33]. What is pacing function? The goal of the pacing function is to determine the size of the training dataset when we train the model. From this set, training batches are

sampled uniformly. According to the research paper "**When do curricula work?**" that we cited in the thesis, the authors adopt 6 different function families to do the experiment, such as the root, logarithmic, linear, quadratic, step and exponential function families. The pacing functions are primarily determined by the two disparate parameters, in this section, we will use the coefficient a and b to represent these two parameters. In the pacing functions, to reach the size of the full training set, parameter a stands for the fractions of training required for the pacing function, parameter b represents how many fractions of the training set would be used at the beginning of the training [17]. Depending on the selection of the parameters of the pacing functions, it is faster for the models to learn harder images if the pair of parameters (a and b) are optimal values.

In our experiment, we ask the "**When do curricula work?**" paper authors for all the pair of parameters on different models, and through these combinations to test the baseline model and curriculum model, and we select the following optimal parameter in the end [Figure 3.5].

	parameter a	parameter b	function
Baseline Model	1.6	0.2	quad
Curriculum Model	1.6	0.0025	root

Figure 3.5: Pacing Function Optimal Parameters.

Chapter 4

Methodology

In the Methodology chapter, in the first place, we will discuss the datasets and environment that we will use in the experiment. Besides, we will define the structure of the baseline model and curriculum model that we will train in the experiments, the both models will adopt the neural network and hyperparamters (optimal) that we mentioned earlier in the Background chapter. Last but not least, we will explain the two experimental procedures, and provide the label tables and flowcharts to make the readers more understand the each steps in two experiments.

4.1 Datasets

4.1.1 CIFAR10

The CIFAR10 dataset [Figure 4.1] [42] are composed of 60000 colour RGB images with each size 32∗32∗3, the images in each class total 6000 (10 classes). Moreover, the whole dataset will be divided into 6 pieces, the training dataset includes five-sixths of the whole dataset, the testing dataset includes one-sixth of the whole dataset. That is to say, the training dataset and the testing dataset will contain the 50000 images and the 10000 images individually, and the training dataset of each class contains 5000 images and the testing dataset of each class contains 1000 images [43].

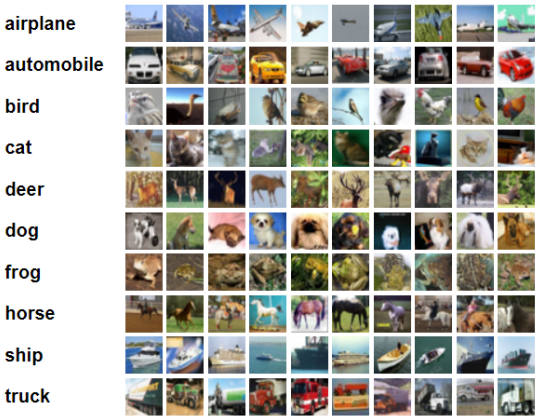

Figure 4.1: 10 Classes Images Sample of the CIFAR10 Dataset.

4.1.2 CIFAR100

The CIFAR100 dataset [Figure 4.2] [44] are composed of 60000 colour RGB images with each size 32∗32∗3, compared with the CIFAR10 dataset, the CIFAR100 dataset includes 100 classes, therefore, the total images in each class is 600. Moreover, the whole dataset will be divided into 6 pieces, the training dataset includes five-sixths of the whole dataset, the testing dataset includes one-sixth of the whole dataset. That is to say, the training dataset and the testing dataset will contain the 50000 images and the 10000 images individually, and the training dataset of each class contains 500 images and the testing dataset of each class contains 100 images [43].

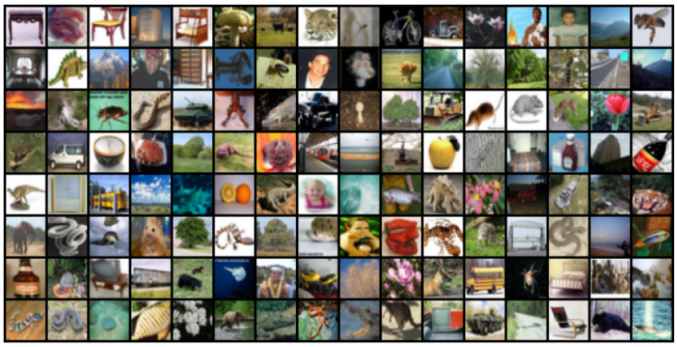

Figure 4.2: 100 Classes Images Sample of the CIFAR100 Dataset.

4.2 Experiment Environment

During the experiment, our coding experimental environment can be divided into two parts. The first environment is to train the baseline model and curriculum model on the UCL server system, owing to the server can save many training time (run faster) and reduce the capacity of the CPU in the computer, the packages and packages version required to train primarily include the following components. The second environment is at the local, we will utilize the jupyter notebook to analyze whole trained models and experimental results, and provide the meaningful plot results at last. If the below packages version do not consistent, it may have an impact on the accuracy of the models or bring about the severe deviations in the prediction.

- torch == 1.4.0
- torchvision == 0.5.0
- scipy, numpy, wget

4.3 Training Models

4.3.1 Baseline Model

In the experiment, the neural network structure and the parameters of the baseline model will be defined as follows.

- ResNet50 neural network.
- SGD optimizer.
- Cross entropy loss function.
- Pacing function, a = 1.6, b = 0.2, quadratic.
- Cosine learning rate decay.
- Batch−size = 128.
- The data in the training procedure are random.

4.3.2 Curriculum Model

In the experiment, the neural network structure and the parameters of the curriculum model will be defined as follows.

- ResNet50 neural network.
- SGD optimizer.
- Cross entropy loss function.
- Pacing function, a = 1,6, b = 0.0025, root.
- Cosine learning rate decay.
- Batch−size = 128.
- The data in the training procedure are from simple to complicated.

4.4 First Experiment - Test Accuracy

In the First Experiment − Test Accuracy section, we will furnish the readers with a brief summary of the different processes in the first experiment and provide a flowchart to exhibit the procedures of this experiment. The first experiment in our thesis will be divided into two small parts, the first part will focus on testing the models with CIFAR10 dataset and the second part will focus on testing the models with CIFAR100 dataset.

4.4.1 In Domain (CIFAR10)

In the first part of the experiment, we will define the training time (epochs) equal to **5 times** at the beginning. After that, we will use the CIFAR10 dataset to train the baseline model and curriculum model with the 5 epochs condition. Meanwhile, the label numbers of each class will correspond to the number 0 to 9 listed in the below CIFAR10 dataset table [Figure 4.3]. After the both models finish training procedure, we will use the CIFAR10 dataset to test the baseline model and curriculum model individually [Figure 4.4], obtain the results of **accuracy**, and further use these results to create the **confusion matrices**, subtract with each other (difference), and comment on what phenomenon exists in the confusion matrix in the end. [**Note:** Each class will contain 1000 testing images.]

Label Names	Label Numbers	Label Names	Label Numbers
airplane	0	dog	5
automobile	1	frog	6
bird	2	horse	7
cat	3	ship	8
deer	4	truck	9

Figure 4.3: CIFAR10 Dataset Label Numbers.

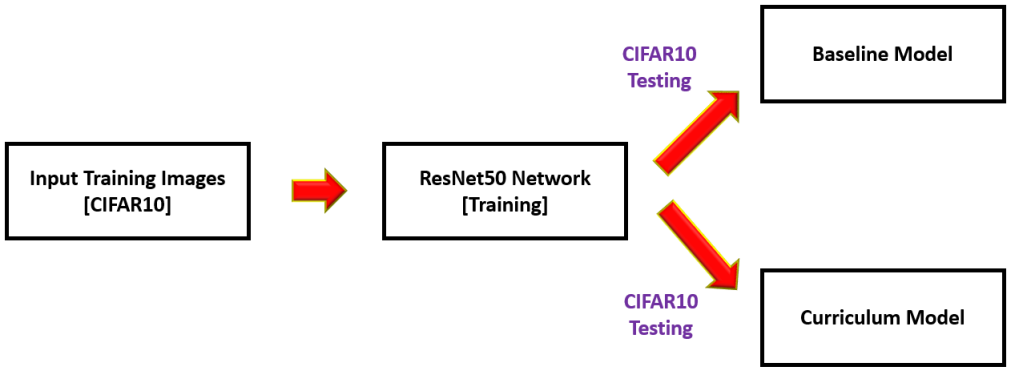

Figure 4.4: CIFAR10 Dataset Flowchart.

4.4.2 Out of Domain (CIFAR100)

In the second part of the experiment, we will define the training time (epochs) equal to **5 times** at the beginning. After that, we will use the CIFAR10 dataset to train the baseline model and curriculum model with the 5 epochs condition. After the both models finish training procedure, we will use the CIFAR100 dataset to test the baseline model and curriculum model individually [Figure 4.6]. However, thanks to the trained models only contain 10 outputs, if we want to test the CIFAR100 dataset in this section, we need to define the same number of classes that matching with the outputs of the trained models (10 classes). Moreover, we will consider that whether both classes are similar and further define the correlations table (matching classes) by ourselves. To begin with, we will choose 10 classes from the CIFAR100 dataset, which are rocket, pickup truck, butterfly, leopard, cattle, wolf, lizard, camel, tank and bus classes. In addition, we will redefine the label numbers of these classes that will correspond to the number 0 to 9 listed in the below CIFAR100 dataset table [Figure 4.5]. After the both models finish testing procedure, we will obtain the results of **accuracy**, and further use these results to create the **confusion matrices**, subtract with each other (difference), and comment on what phenomenon exists in the confusion matrix in the end. [**Note:** We choose 10 classes from the CIFAR100 dataset, therefore, each class will contain 100 testing images.]

Label Names	Label Numbers	Label Names	Label Numbers
rocket	0	wolf	5
pickup truck	1	lizard	6
butterfly	2	camel	7
leopard	3	tank	8
cattle	4	bus	9

Note:
We will choose the 10 classes from the CIFAR100 dataset what we want, and further redefine the label numbers that correspond to the number 0 to 9 individually.

Figure 4.5: CIFAR100 Dataset Label Numbers

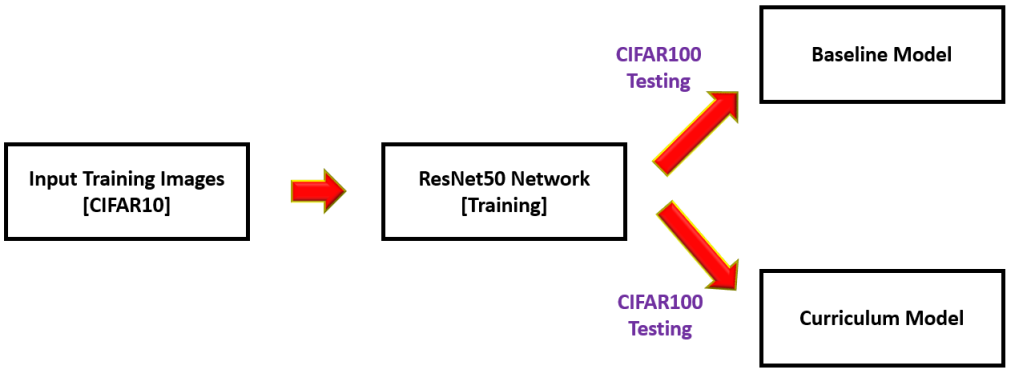

Figure 4.6: CIFAR100 Dataset Flowchart.

4.5 Second Experiment - Ranking Misclassified

In the Second Experiment − Ranking Misclassified section, we will elaborate on the every procedures required for the second experiment. On the basis of the two small experiments mentioned in the above In Domain (CIFAR10) and Out Domain (CIFAR100) sections, we will extend both experiments to do more investigations. The main target in the second experiment will focus on the misclassified images [45], we will implement some steps to find out the images which possess much deviation between the predict and answer (3 largest misclassified images) of each class. According to the misclassified images, we can easily understand that the prediction preference of the baseline model and curriculum model.

4.5.1 In Domain (CIFAR10)

In this section, we will extend the In Domain (CIFAR10) experiment mentioned at the First Experiment − Test Accuracy section. How to find the 3 largest misclassified images of each class? We will illustrate the processes more detailed as follows. During the testing procedure, when we input each image, we will use the cross−entropy loss function, cooperating with the softmax approach to record the probability distribution of this predicted image. As a result of we will choose the top 1 accuracy of each image, therefore, in line with the probability distribution, we will select the largest probability and find its corresponding label to be the predicted label. When the predicted label and the answer label are not matched with each other, we will through the probability distribution of each image, calculate the absolute value between the predicted label and the answer label (subtract their probabilities), and further store this value in the **list** of each answer class (python list with 10 classes sub−list, each sub−list includes 3 positions). Besides, we will also use the same type of the python list to store the value's probability distribution. During the calculation procedures, we will use a update function to record the misclassified value and probability distribution of each image. When the input misclassified value is larger than any position of the current 3 position values, we will sort the both lists again and change them to the

new order, however, when the input misclassified value is smaller than all positions of the current 3 position values, we will not do anything. After both models finish testing procedure, we will plot the 30 misclassified images and their probability distribution following the form 3∗10, where each column represents that each class with 3 largest misclassified images, and further use these results to do more comments and explanations in the end. [**Note:** We do not store all the misclassified values in the testing procedure, we only record the current 3 largest values, because, it can make the testing procedure become more faster and reduce the capacity of the CPU in the computer.]

4.5.2 Out of Domain (CIFAR100)

In this section, we will extend the Out of Domain (CIFAR100) experiment mentioned at the First Experiment − Test Accuracy section. All of the ranking misclassified experiment processes will same with the In Domain (CIFAR10) we mentioned above. After the baseline model and curriculum model finish testing procedure, we will plot the 30 misclassified images and their probability distribution following the form 3∗10, where each column represents that each class with 3 largest misclassified images, and further use these results to do more comments and explanations in the end.

Chapter 5

Results

The purpose of the Results chapter is to present all of the results we obtained from the Test Accuracy and Ranking Misclassified experiments, analyze them and judge the reasons why these results will happen or occur, and further provide the comprehension conclusions. First of all, we will provide the loss curve and accuracy curve results of the baseline model and curriculum model, using the CIFAR10 dataset to train both models with 5 epochs condition. Furthermore, we will exhibit the test accuracy results and the confusion matrices results of the Test Accuracy → In Domain (CIFAR10) and Out of Domain (CIFAR100) sections. Last but not least, we will display the all the misclassified images of the Ranking Misclassified → In Domain (CIFAR10) and Out of Domain (CIFAR100) sections.

5.1 Loss Curve and Accuracy Curve

In this section, we will provide the loss curve and accuracy curve of the baseline model and curriculum model after both models finish training procedure with the limited training time condition (5 epochs).

5.1.1 Baseline Model

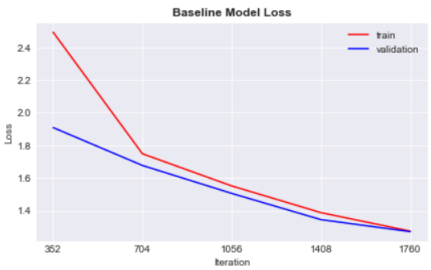

Figure 5.1: Baseline Model Loss Curve.

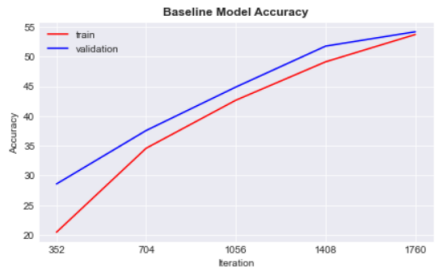

Figure 5.2: Baseline Model Accuracy Curve.

5.1.2 Curriculum Model

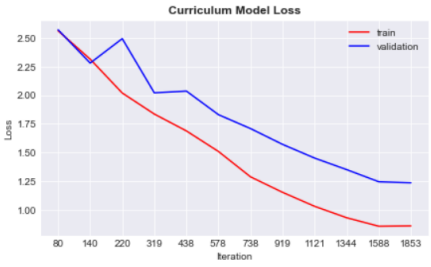

Figure 5.3: Curriculum Model Loss Curve.

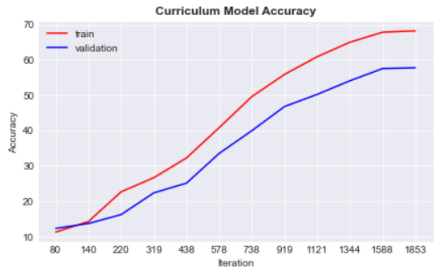

Figure 5.4: Curriculum Model Accuracy Curve.

5.2 First Experiment - Test Accuracy Results

In the First Experiment – Test Accuracy Results section, we will provide the accuracy table results [Figure 5.5] of the baseline model and curriculum model of the In Domain (CIFAR10) and Out of Domain (CIFAR100) sections respectively.

5.2.1 Test Accuracy

In Domain (CIFAR10)

After the baseline model and the curriculum model finish the testing procedure, we will obtain the test accuracy results that are **54.45**% and **57.81**% respectively.

Out of Domain (CIFAR100)

After the baseline model and the curriculum model finish the testing procedure, we will obtain the test accuracy results that are **32.03**% and **32.81**% respectively.

	Baseline Model	Curriculum Model
CIFAR 10 test dataset	54.45%	57.81%
CIFAR 100 test dataset	32.03%	32.81%

Figure 5.5: Test Accuracy Table.

Conclusion

In the light of the Test Accuracy Table results, we can notice that when we use the CIFAR10 dataset to test the baseline model and curriculum model (In Domain), the curriculum model will possess the better performance, which is the same results in the paper "**When do curricula work?**". Moreover, we also can notice that when we use the CIFAR100 dataset to test the baseline model and curriculum model (Out of Domain), the curriculum model will also contain the better performance (slight). In other words, we can conclude that when we limit the training time (5 epochs) on the models during the training process, even though we utilize the different dataset to test the models, the curriculum learning will also work (generalization) under this circumstance.

5.2.2 Confusion Matrix

In the machine learning fields, what type of approach can be implemented to evaluate the classification model performance? This indicator is called the confusion matrix. With the confusion matrix, we are not only able to observe the model's performance in each class, but we also can calculate the accuracy that corresponds to each class. On top of that, the confusion matrix can assist people to identify which classes are not readily distinguishable. To cite an instance, such as how many of the airplane images are classified into the bird images, therefore, we can through design various features to make the classes more noticeable [46].

In Domain (CIFAR10)

After the baseline model and the curriculum model finish the testing procedure, we can compute the numbers of mistake prediction of each class, and further plot the confusion matrix at last. In the plot results, the x−axis represents the predicted classes and the y−axis represents the answer classes. For the purpose of making the results uncomplicated for us to observe, we will compute the difference [**curriculum model − baseline model**] among each cell in the confusion matrix, and further obtain the following difference confusion matrix [Figure 5.6] in the end.

Owing to we use the curriculum model to subtract the baseline model, therefore, we can understand that if the difference value in the cell is positive, it represents that the curriculum model will be more frequent in this combination of the predict and answer classes, whereas, if the difference value in the cell is negative, it represents that the baseline model will be more frequent in this combination of the predict and answer classes.

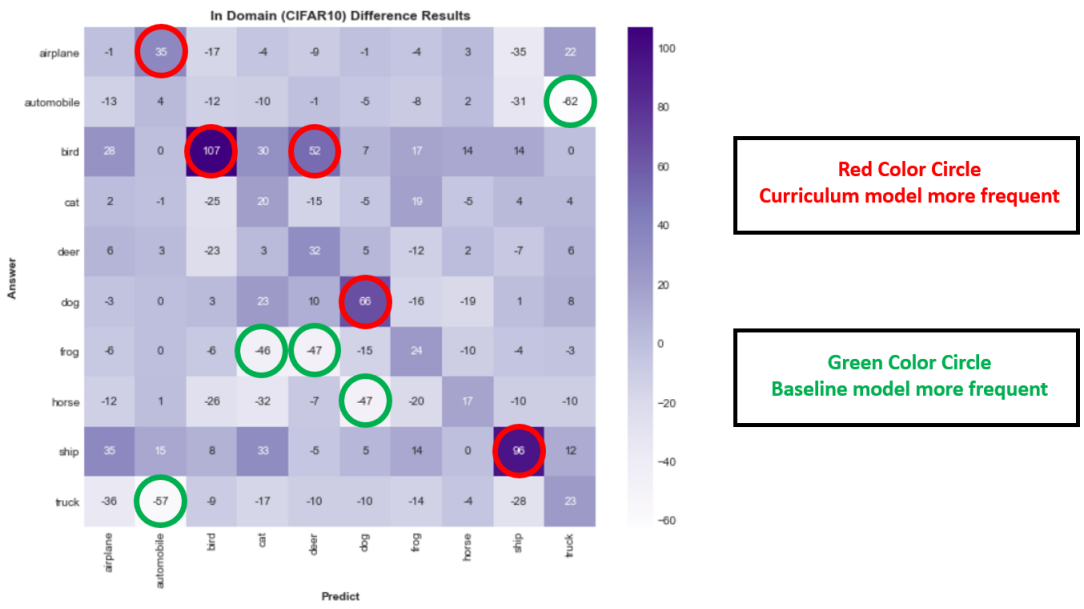

Figure 5.6: In Domain (CIFAR10) Difference Confusion Matrix.

Conclusion

In the In Domain (CIFAR10) difference confusion matrix results, we will choose the 5 larger value cells and 5 smaller value cells to analyze and investigate.

On the basis of the red circles that we select in the confusion matrix, we can notice that three of the red circles are on the diagonal, which are representative of combinations bird, dog and ship classes individually. In other words, when we use the dataset to test the curriculum model, in these three classes, the curriculum model can predict more accurate in contrast to the baseline model (diagonal values are larger enough). The other two red circles are representative of the deer class predicted as bird class and automobile class predicted as airplane class quite frequent in comparison to the baseline model.

According to the green circles that we choose in the confusion matrix, we can find that they are representative of the automobile class predicted as truck class, cat class predicted as frog class, deer class predicted as frog class, dog class predicted as horse class and truck class predicted as automobile class quite frequent in comparison to the curriculum model.

What may lead to these mistake predictions? In our opinion, there may include a lots of possible reasons. One possible reason is that because of the predict object image does not exhibit the complete form, such as the image only includes the facial or the body features, therefore, it may bring about the models catch the error features and make a mistake prediction. Another possible reason is that the predict image is very formidable to distinguish or the shape of the object is very similar, therefore, it may also cause the models to make a mistake prediction.

Out of Domain (CIFAR100)

After the baseline model and the curriculum model finish the testing procedure, we will do the same steps as we mentioned in the above In Domain (CIFAR10) section. Ultimately, we will obtain the following difference confusion matrix [Figure 5.7].

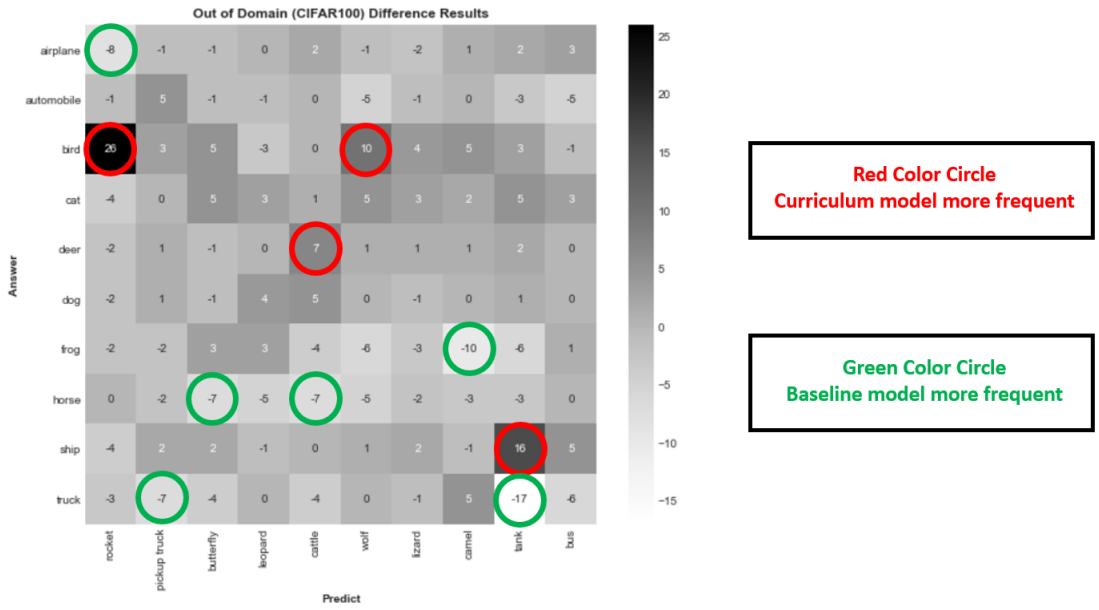

Figure 5.7: Out of Domain (CIFAR100) Difference Confusion Matrix.

Conclusion

In the Out of Domain (CIFAR100) difference confusion matrix results, we will choose the cell value greater or equal to 7 or the cell value less than or equal to -7 to analyze and investigate.

On the basis of the red circles that we choose in the confusion matrix, we can find that they are representative of the rocket class predicted as bird class, cattle class predicted as deer class, wolf class predicted as bird class and tank class predicted as ship class quite frequent in comparison to the baseline model.

According to the green circles that we choose in the confusion matrix, we can find that they are representative of the rocket class predicted as airplane class, pickup truck class predicted as truck class, butterfly class predicted as horse class, cattle class predicted as horse class, camel class predicted as frog class and tank class predicted as truck class quite frequent in comparison to the curriculum model.

What may lead to these mistake predictions? In our opinion, there may include a lots of possible reasons. In the first place, such as the predict object image does not exhibit the complete form or the image is very formidable to distinguish are all the possible reasons will cause the models to make a mistake prediction. Besides, the other possible reason is as a result of we use

the different dataset to test the models, the correlations of each class may not be so relevant (similar), therefore, it will also cause the models to make a mistake prediction.

What's the slight advantage of the curriculum model in contrast with the baseline model? We can notice that the confusion matrix exists an interesting classification result, which is the rocket class predicted as the bird class quite often in the curriculum model. From our point of view, we assume that the curriculum model may take shortcuts, such as using the background to predict the image result, therefore, it will have the higher possibility to predict the rocket image as the bird image.

5.3 Second Experiment - Ranking Misclassified Results

In the Second Experiment − Ranking Misclassified Results section, we will provide 30 misclassified images and probability distribution results of the baseline model and curriculum model of the In Domain (CIFAR10) [Figure 5.8] [Figure 5.9] and Out of Domain (CIFAR100) [Figure 5.10] [Figure 5.11] sections respectively. [**Note:** The 30 misclassified images and their probability distribution will follow the form 3∗10, where each column represents that each class with 3 largest misclassified images.]

5.3.1 Misclassified Images

In the probability distribution results, we will use the different colors to stand for the different definitions. The red and green colors are representative of the current class (answer) and highest probability in the probability distribution (error prediction). The purple color is representative of the other classes.

In Domain (CIFAR10)

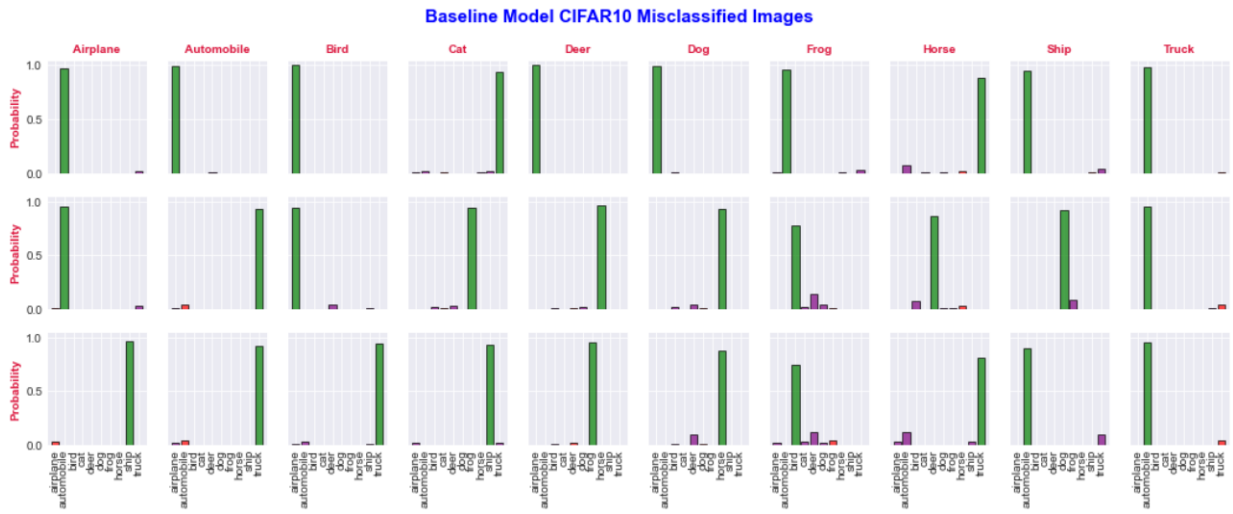

Figure 5.8: Baseline Model 30 Misclassified Images and Probability Distribution.

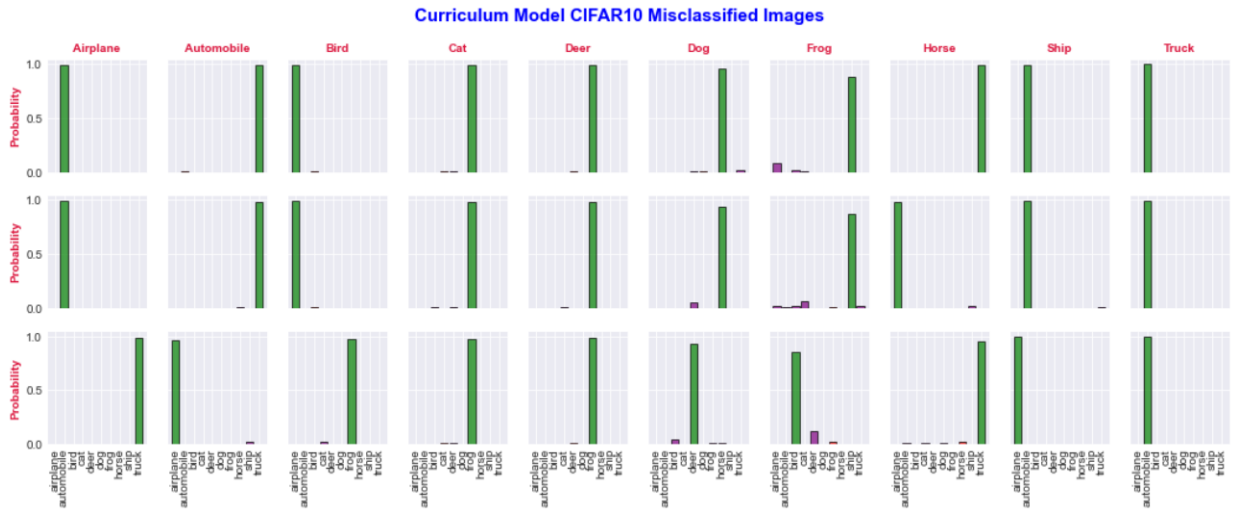

Figure 5.9: Curriculum Model 30 Misclassified Images and Probability Distribution.

Conclusion

On the basis of the above baseline model and curriculum model misclassified images results, we can find that there exist some phenomena and trends in the results. To begin with, we can observe that the 3 largest misclassified images of the truck class in the baseline model and curriculum model are predicted to the same mistake classes sequence. Apart from this phenomenon, we also can notice that, in the curriculum model, the 3 largest misclassified images of cat class, deer class and truck class are all predicted to the identical mistake classes individually (at the each class). In addition, we can through the overall trend of the 30 misclassified images in the baseline model and curriculum model to understand that in comparison to the baseline model, the curriculum model tends to be more confident about mistake in domain (mistake prediction incline to specific class).

Out of Domain (CIFAR100)

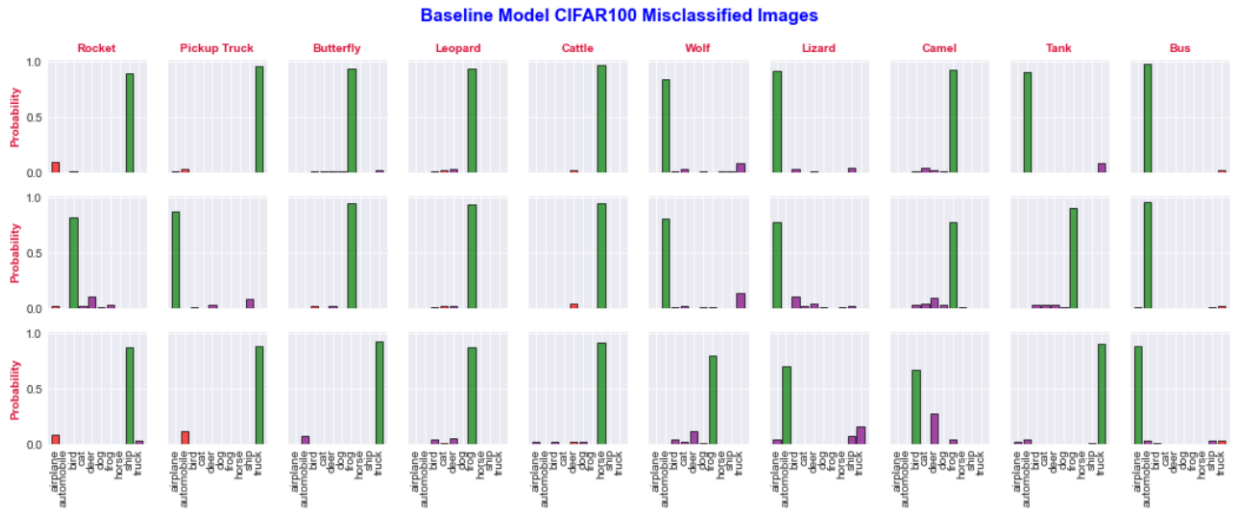

Figure 5.10: Baseline Model 30 Misclassified Images and Probability Distribution.

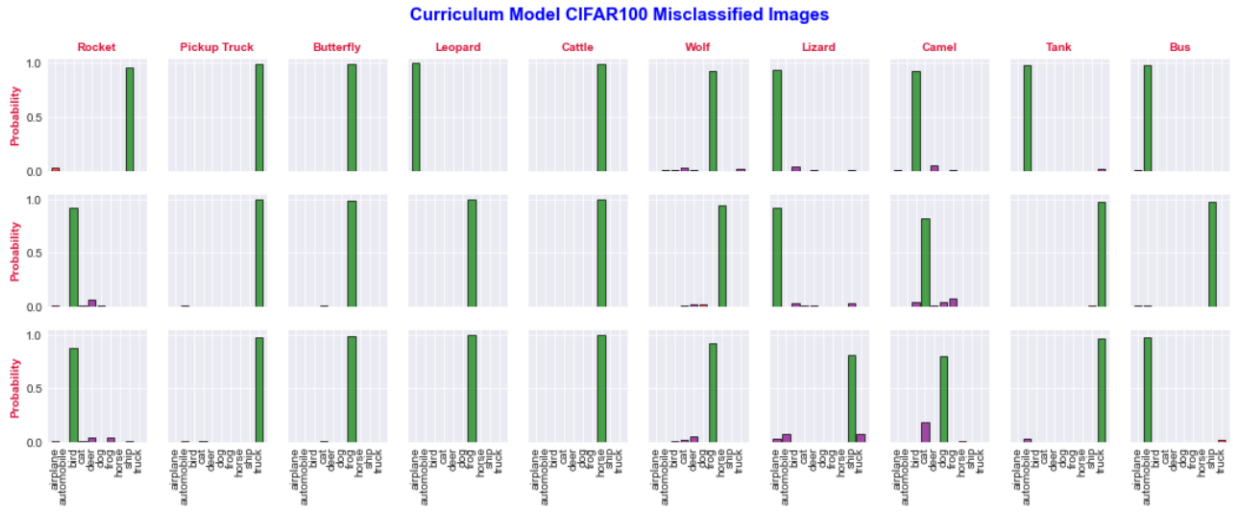

Figure 5.11: Curriculum Model 30 Misclassified Images and Probability Distribution.

Conclusion

In line with the above baseline model and curriculum model misclassified images results, we can find that there exist some phenomena and trends in the results. In the first place, we can observe that the 3 largest misclassified images of the cattle class in the baseline model and curriculum model are predicted to the same mistake classes sequence. Aside from this phenomenon, we also can notice that, in the curriculum model, the 3 largest misclassified images of pickup truck class, butterfly class and cattle class are all predicted to the identical mistake classes individually (at the each class). Furthermore, we can through the overall trend of the 30 misclassified images in the baseline model and curriculum model to understand that in comparison to the baseline model, the curriculum model tends to be more confident about mistake out of domain (mistake prediction incline to specific class).

Chapter 6

Discussion and Future Directions

In this thesis, all of the experiments will based on the two models: baseline model and curriculum model, and two datasets: CIFAR10 and CIFAR100. The neural network structure of both models are ResNet50. We will use both models and their corresponding optimal hyperparameters to train the CIFAR10 dataset with 5 epochs (limited training time) individually, and further use the CIFAR10 and CIFAR100 datasets to test both models individually. The main difference between two models will come from the order of the training data, the data input to the baseline model are random, and the data input to the curriculum model are from simple to complicated. After both models finish testing procedure, we will obtain the accuracy **54.45**% and **57.81**% with CIFAR10 testing dataset, and **32.03**% and **32.81**% with CIFAR100 testing dataset. In line with the experimental results, we can understand that curriculum learning will work in both cases. Hence, we can further conclude that the curriculum learning will work when we use the different dataset to test the model (generalization) under this circumstance.

This thesis contains the following contributions: First of all, through the difference confusion matrix (In Domain (CIFAR10) and Out of Domain (CIFAR100)), we investigate the type of errors on the results, and further make assumptions that the curriculum model may take shortcuts when predict the images and the definiton of the correlation table may have an influence on the accuracy results. Besides, we can use the test accuracy table of both models to further realize that when we use the testing dataset that is out of domain, in our experiments, the curriculum learning also will work. Eventually, through the 3 largest misclassified images, we can further understand what are the prediction properties of these two models.

Even though we obtain the good achievements in the experiments, we still encounter many difficulties during the experimental processes. In the first place, in the case of experimental environment settings, if the packages version in the experiments are different from the version in the paper "**When do curricula work?**", the accuracy of the calculation will be very poor, and further bring about the incorrect experimental results. Besides, how to set a good epoch hyperparameter is a significant problem. In the experiments, we define the 5 epochs to train both models. According to the test accuracy results (Out of Domain), even though we still can realize that the curriculum learning will work when the testing dataset is out of domain (CIFAR100), we cannot see the evident accuracy results of baseline model (**32.03**%) and curriculum model (**32.81**%), the accuracy results are very similar (nearly) with each other. However, the number

of epochs cannot be determined too large, because, when the number of training times is large enough, the training data order of the curriculum learning approach and random approach will be very identical. In addition, we only train both models once, and further use them to make the predictions. However, this approach may lead to the results do not plausible enough. Hence, we should train both models a lots of times, calculate the average or standard deviation values, and through in comparison with these values and further choose the optimal models to do the following testing experiments.

From our point of view, the possible research directions in the future may include the following aspects.To begin with, we can follow the idea we mentioned earlier that is related to training the models a lots of times, calculating the average and standard deviation values, and further through these values to choose the optimal models to implement the following experiments. Additionally, we should define the limited training time with various values, such as 10, 15, 20 epochs and so on. After the both models finish training procedure, using the different testing dataset (out of domain) to test the models, and through the test accuracy table to comment whether the curriculum learning will also work with the various epoch values (gradually increase, testing out of domain). In fact, in the paper "**When do curricula work?**" that we cited have mentioned an interesting phenomenon. The authors notice that when they set the epoch value is equal to 50, the effectiveness of the curriculum learning will not be conspicuous (in domain). Furthermore, in the light of the First Experiment − Test Accuracy Results Out of Domain difference confusion matrix, we hypothesize that the curriculum model may take shortcuts when predict the image results, such as using the background to predict the image results, so that the rocket images are quite often predicted to the bird class. Hence, we can design a spurious correlation table for each class, add the possible background of each class and further train the models. When the models during the testing procedure, using the background to predict the image results can exclude the class that does not belong to environments or regions, and further improve the accuracy and enhance the models performance. Last but not least, we can also through the current misclassified images to make a extension. We can restore the original images of these misclassified images, in addition, we can also find the 3 smallest misclassified images, compare with these new results and current results, and further analyze whether the results exist any correlation or trend. Due to the time constraints, we do not have much time to extend the experiments, otherwise, these are some interesting viewpoints that we really want to take time to explore and investigate.

Bibliography

[1] Judith Avrahami, Yaakov Kareev, Yonatan Bogot, Ruth Caspi, Salomka Dunaevsky, and Sharon Lerner. "Teaching by examples: Implications for the process of category acquisition." The Quarterly Journal of Experimental Psychology Section A, 50(3):586–606, 1997.

[2] Jeffrey L Elman. "Learning and development in neural networks: The importance of starting small." Cognition, 48(1):71–99, 1993.

[3] Terence D Sanger. "Neural network learning control of robot manipulators using gradually increasing task difficulty." IEEE transactions on Robotics and Automation, 10(3):323–333, 1994.

[4] Yoshua Bengio, Jerôme Louradour, Ronan Collobert, Jason Weston. "Curriculum learning." In Proceedings of International Conference on Machine Learning, 2009.

[5] Lilian Weng. "Curriculum for Reinforcement Learning." Jan 29, 2020.

[6] Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. "Curriculum learning of multiple tasks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5492–5500, 2015.

[7] Nikolaos Sarafianos, Theodore Giannakopoulos, Christophoros Nikou, and Ioannis A Kakadiaris. "Curriculum learning for multi-task classification of visual attributes." In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 2608–2615, 2017.

[8] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. "Curriculumnet: Weakly supervised learning from large-scale web images." In Proceedings of the European Conference on Computer Vision (ECCV), September 2018.

[9] Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. "Dynamic curriculum learning for imbalanced data classification." In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.

[10] Wojciech Zaremba and Ilya Sutskever. "Learning to execute." arXiv preprint arXiv:1410.4615, 2014.

[11] Qian Long, Zihan Zhou, Abhibav Gupta, Fei Fang, Yi Wu, Xiaolong Wang. "Evolutionary Population Curriculum for Scaling Multi-Agent Reinforcement Learning." ICLR 2020.

[12] Kun Shao, Yuanheng Zhu, Dongbin Zhao. "StarCraft Micromanagement with Reinforcement Learning and Curriculum Transfer Learning." 3 Apr 2018.

[13] Volkan Cirik, Eduard Hovy, and Louis-Philippe Morency. "Visualizing and understanding curriculum learning for long short-term memory networks." arXiv preprint arXiv:1611.06204, 2016.

[14] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. "Competence-based Curriculum Learning for Neural Machine Translation." In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 1162–1172, 2019.

[15] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165, 2020.

[16] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683, 2019.

[17] Xiaoxia Wu, Ethan Dyer, Behnam Neyshabur. "WHEN DO CURRICULA WORK?" Published as a conference paper at ICLR 2021.

[18] Urwa Muaz. "Case Study: Is Spurious Correlations the reason why Neural Networks fail on unseen data?" Apr 18, 2021.

[19] Jerry Wei, Arief Suriawinata, Bing Ren, Xiaoying Liu, Mikhail Lisovsky, Louis Vaickus, Charles Brown, Michael Baker, Mustafa Nasir-Moin1, Naofumi Tomita, Lorenzo Torresani, Jason Wei1 and Saeed Hassanpour. "Learn like a Pathologist: Curriculum Learning by Annotator Agreement for Histopathology Image Classification." 29 Sep 2020.

[20] Mihail Burduja, Radu Tudor Ionescu. "Unsupervised Medical Image Alignment With Curriculum Learning." 20 Feb 2021.

[21] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do and Kaori Togashi. "Convolutional neural networks: an overview and application in radiology." 22 June 2018.

[22] Kaiqing Zhang, Zhuoran Yang, Tamer Başar. "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms." 24 Nov 2019.

[23] Diksha Khurana, Aditya Koli, Kiran Khatter and Sukhdev Singh. "Natural Language Processing: State of The Art, Current Trends and Challenges." August 2017.

[24] Ralf C. Staudemeyer, Eric Rothstein Morris. "Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks." 12 Sep 2019.

[25] Nal Kalchbrenner and Phil Blunsom. "Recurrent continuous translation models." In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1700–1709.

[26] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate." 2015. In International Conference on Learning Representations.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016

[28] Karen Simonyan, Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, 2014.

[29] Sergey Zagoruyko, Nikos Komodakis. "Wide residual networks." In Proceedings of the British Machine Vision Conference (BMVC), pp. 87.1–87.12. BMVA Press, September 2016.

[30] Mingxing Tan, Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International Conference on Machine Learning, pp. 6105–6114, 2019.

[31] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, Kilian Q Weinberger. "Densely connected convolutional networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708, 2017.

[32] Diederik P Kingma, Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.

[33] Guy Hacohen and Daphna Weinshall. "On the power of curriculum learning in training deep networks." ICML, 2019.

[34] MURAT H. SAZLI. "A BRIEF REVIEW OF FEED-FORWARD NEURAL NETWORKS." January 2006.

[35] Baki Er. "Microsoft Presents : Deep Residual Networks." Aug 10, 2016.

[36] Qingge Ji, Jie Huang, Wenjie He and Yankui Sun. "Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images." February 2019, Algorithms 12(3):51.

[37] Ryan Lu. "Learning Model: Gradient Descent -Learning rate - Optimizer introduction and selection." Mar 25, 2019.

[38] Daniel Ramos, Javier Franco-Pedroso, Alicia Lozano-Diez and Joaquin Gonzalez-Rodriguez. "Learning Model: Gradient Descent -Learning rate - Optimizer introduction and selection." 20 March 2018.

[39] D. Randall Wilson, Tony R. Martinez. "The Need for Small Learning Rateson Large Problems." In Proceedings of the 2001 International Joint Conference on Neural Networks (IJCNN'01), 115-119.

[40] Ilya Loshchilov, Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts." 13 Aug 2016.

[41] Ken Huang. "Epoch, Batch size, Iteration, Learning Rate." Nov 6, 2020.

[42] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "The CIFAR-10 dataset."

[43] From Wikipedia, the free encyclopedia. "CIFAR-10." 29 March 2021.

[44] Ali Tabak. "Image Classification in a Nutshell: 5 Different Modelling Approaches in PyTorch with CIFAR100." Jul 4, 2020.

[45] Jorge Marco Blanco, Luis Baumela Molina. "Detection of Misclassified and Adversarial examples in Deep." 2019.

[46] Sofia Visa, Brian Ramsay, Anca Ralescu, Esther van der Knaap. "Confusion Matrix-based Feature Selection." January 2011.