Learning to Recognize with few Examples

(Few-Shot with Meta-Learning)

Author

SHEIKH ADEEL AHMED

Regn # 00000206173

Supervisor

DR HASAN SAJID

DEPARTMENT OF ROBOTICS AND ARTIFICIAL INTELLIGENCE

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

AUGUEST, 2021

Learning to Recognize with few Examples

(Few-Shot with Meta-Learning)

Author

SHEIKH ADEEL AHMED

Regn # 00000206173

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machines Engineering

Thesis Supervisor:

Dr. Hassan Sajid

Thesis Supervisor's Signature: ____

DEPARTMENT OF ROBOTICS AND ARTIFICIAL INTELLIGENCE

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

AUGUST 2021

National University of Sciences and Technology

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: **Mr. Shiekh Adeel Ahmed Regn#00000206173** Titled: **"Learning to Recognize with few Examples"** be accepted in partial fulfillment of the requirements for the award of **MS Robotics & Intelligent Machine Engineering** degree. **(Grade_____)**

Examination Committee Members

1. Name: Dr. Hasan Sajid Signature:________________

2. Name: Dr. Muhammad Jawad Khan Signature:________________

3. Name: Dr. Karam Dad Kallu Signature:________________

Supervisor's name: Dr. Hasan Sajid Signature: ________________

Date:____________________

____________________ ____________________

Head of Department Date

COUNTERSIGNED

Date:___________ _______________

Principal

Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by *Sheikh Adeel Ahmed* (Registration No. 00000206173), of Department of Robotics and Intelligent Machine Engineering (SMME) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free from plagiarism, errors and mistakes and is accepted as a partial fulfilment for award of MS Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature: ____________________

Name of Supervisor: Dr. Hasan Sajid

Date: ____________________

Signature (HOD): ____________________

Date: ____________________

Signature (Principal): ____________________

Date: ____________________

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

SHEIKH ADEEL AHMED

Regn # 00000206173

Dr. HASSAN SAJID

(Supervisor)

Declaration

I certify that this research work titled "*Learning to Recognize with few Examples"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

SHEIKH ADEEL AHMED

Reg# 00000206173

Copyright Statement

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of NUST School of Mechanical & Manufacturing Engineering (SMME). Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST School of Mechanical & Manufacturing Engineering, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the SMME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST School of Mechanical & Manufacturing Engineering, Islamabad.

Acknowledgements

My humble gratitude is for the Allah, the most beneficent and merciful, who gave me strength to stand against all the odds and passed through difficult barriers along the way to reach till this point. So, I thank Him for his guidance and presence in my heart. It is by Allah's blessings that I was bestowed by such parents who stood by me, siblings whose tireless support gave me strength during my project and thesis. My heartly regard is for my supervisor Sir Hasan Sajid for his guidance throughout the tenure of my thesis.

I pay my special regards to members of my GEC committee, Dr. Muhammad Jawad Khan, and Dr Karam Daad.

I would like to thank my fellows and co-workers who succored me in different aspects of my research, and never let me face the failure at any stage of my research.

Eventually I pay my humble thanks to all those who stood by me, bore my lows, and became the silver lining at my worst and helped me grow through my work.

"I am grateful of you all and your presence in my life"

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

Few samples learning (FSL) is significant and challenging in the field of machine learning. The main challenge of few-shot learning is the deficiency of samples. Training on much smaller training sets while maintaining nearly the same accuracy would be very beneficial. Meta-learning is the process of learning how to learn. It is a subfield of machine learning where automatic learning algorithms are applied to metadata about machine learning experiment. Reptile is the application of the Shortest Descent algorithm to the meta-learning setting. We have developed a simple meta-learning algorithm named stomatopods inspired from Reptile which works by repeatedly sampling a task, performing stochastic gradient descent on it, and updating the initial parameters towards the final parameters learned on that task. The obtained results show a significant improvement in accuracies on four different datasets and found that the results were better.

Key Words: *Meta Learning, Few-Shot Learning, Triplet Network, Convolution Neural Network*

Table of Contents

FORM TH-4 .... **i**

Thesis Acceptance Certificate .... **ii**

Plagiarism Certificate (Turnitin Report) .... **iii**

Declaration .... **iv**

Copyright Statement .... **v**

Acknowledgements .... **vi**

Abstract .... **viii**

Table of Contents .... **ix**

List of Figures .... **xi**

List of Tables .... **xiii**

List of Abbreviations .... **xiv**

CHAPTER 1 INTRODUCTION **15**

1.1 Background 15

1.2 Problem Statement 16

1.3 Aims and Objectives 16

1.4 Research Methodology 17

1.5 Summary 17

CHAPTER 2 LITERATURE REVIEW **18**

CHAPTER 3 Methodology **21**

3.1 Dataset Selection 21

3.1.1 MNIST 21

3.1.2 Fashion MNIST 22

3.1.3 Cifar-10 23

3.1.4 PCB 24

3.2 Sample Selection for Few-Shot Learning 24

3.3 Objective Function 25

3.3.1 Convolutional Neural Network (CNNs) 25

3.3.2 CNN Training Hyperparameters 27

3.3.3 CNN Accuracy Plots 28

3.3.4 Results ... 28
3.4 Siamese Networks ... 29
3.4.1 Contrastive Loss: ... 30
3.4.2 Triplet Loss ... 31
3.5 Triplet Network ... 32
3.5.1 Triplet Training ... 32
3.5.2 Triplet Network Results ... 33
3.6 Reptile-Meta learning ... 34
3.6.1 Reptile (Training) ... 36
3.6.2 Results ... 37
3.7 Our Approach ... 37
3.7.1 Mini Batches and Data Loader ... 37
3.7.2 Dataset Post processing and Implementation in TFRecord ... 39
3.7.3 Stomatopods Model Definition: ... 40
3.7.4 Model Training ... 43
3.7.5 Pseudocode for updating parameter ... 45
3.7.6 Training of Stomatopods ... 45
CHAPTER 4 Results and Discussion ... 47
4.1 Intra-Class and inter Class variance: ... 48
4.2 Resolution and quality of Data ... 49
CHAPTER 5 Conclusion ... 51
5.1 Conclusion ... 51
REFERENCES ... 52

List of Figures

Figure 1 MNIST dataset images 22

Figure 2: Fashion MNIST dataset images 22

Figure 3: Cifar-10 Dataset Images and class name 23

Figure 4: PCB blood cell Dataset images 24

Figure 5 CNNs Feature Learner cycle 26

Figure 6: Mobile Net Architecture 27

Figure 7 CNN Training Accuracy plots 28

Figure 8 Accuracy Bar CNNs 29

Figure 9 Siamese Network 30

Figure 10: Triplet Example 31

Figure 11: Triplet Network with Triplet Loss 31

Figure 12: Feature Extractor architecture 32

Figure 13: Triplet Network training Accuracies 33

Figure 14: Accuracy Bar of CNN with Siamese 34

Figure 15 Meta-Architecture Model-based Recurrent network 35

Figure 16: Reptile training graph on MNIST dataset 36

Figure 17 Reptile Training graphs 36

Figure 18 Retile comparison accuracy bar chart 37

Figure 19 Mini Batches flowchart 38

Figure 20: Our approach Data loading pipeline 39

Figure 21: Stomatopods Model 41

Figure 22 Stomatopods Architecture details 42

Figure 23 Stomatopod's algorithm learning cycle 44

Figure 24 Stomatopods MNIST training accuracy plot 45

Figure 25 Stomatopods Fashion MNIST training accuracy plot 46

Figure 26 Stomatopods training accuracy plot 46

Figure 27 Our Approach Comparison Accuracy bar chart 47

Figure 28 Illustration of intra class and inter class variation 48

Figure 29 Grayscale dataset accuracy bar chart... 49

Figure 30 Colored dataset accuracy bar chart... 50

List of Tables

Table 2-1: Dataset Description 21

Table 2-2: Modified Dataset details 23

Table 3-1: CNN Training Hyper parameters 27

Table 3-2 : Triplet Hyper parameters 32

List of Abbreviations

Abbreviation	Description
CNN	Conventional Neural Network
DL	Deep Learning
ML	Machine Learning
AI	Artificial Intelligence
STD	Stochastic gradient descent
FSL	Few-Shot Learning
LO	Low-shot Learning
PCA	Principal component analysis
PBC	Public Blood Cell Dataset

CHAPTER 1

INTRODUCTION

1.1 Background

Artificial intelligence is the latest technology that is rapidly evolving and emerging in every field of life. The data and information are entered once and could always be retrieved for any kind of use, and that information is learned either by supervised learning or unsupervised learning. Supervised learning or Unsupervised learning are the two aspects of machine learning where it has introduced a lot of techniques which are dealing with large datasets to provide the solutions of the problems [1]. Such technique especially with supervised learning has returned really good results in term of computer vision, speech recognition and activity detection. Even yet, there were several exceptions or sectors where machine learning struggled to produce satisfactory results because to restrictions or limitations. For example, in machine learning, for almost all the solutions, proper dataset is required that helps in the training the models. The amount of required data can vary as per the problem or requirements. Dealing with large datasets by using the deep learning techniques could be costly in terms of computational cost and time. Human visual systems, on the other hand, can detect new classes with only a handful of instances. The ability to generalize to new classes with a small number of labeled samples for each class is hence of significant importance [2]. As a species, very adept at using the prior knowledge to swiftly absorb new concepts from a small amount of information. Few-shot learning (FSL) is a technique used for machine learning [3]. Few-shot learning methods have been developed to solve the challenge of acquiring new knowledge or ideas from a restricted number of examples [4]. Meta-learning is a promising path for few-shot classification since it extracts and propagates transferable knowledge from a collection of tasks to minimize over fitting and increase generalization by preventing over fitting. Techniques that use model initialization, metric learning methods, and hallucination-based methods are a few examples. Similarly, another study has shown encouraging results by estimating the scores of the classifier for emerging classes [5]. The major goal of the few-shot learning is to predict the unseen image based on few learning samples [6]. Machine learning is evolving over the years and the learning revolves around the data [7]. But sometimes the more data to train the machine does not give accurate models and that's where few-shot learning is used to increase the accuracy of the models [8]. It helps in generalizing the data by avoiding the overfitting and underfitting of the model and helps in the reduction of the expenses [9].

1.2 Problem Statement

In general, deep supervised learning models require a high number of training examples. While this is going on, it appears that humans can generalize fast from a small number of examples. As a means of bridging this gap, getting machines to learn from small data is crucial to the process. Few-shot learning (FSL) is one way to make models more efficient in terms of the number of samples used. If just a few instances of each class are provided, models must learn to distinguish between them. Another type of FSL, dubbed one-shot learning (OSL), has been developed as a result of recent advances in this domain.

1.3 Aims and Objectives

In few studies, researchers have offered a comprehensive analysis of a few-shot categorization issue, which sheds fresh insight on the subject. Comparison studies to examine various typical few-shot categorization techniques on a level playing field, according to our results, adopting a deep backbone reduces the performance gap between different techniques [10]. As a second benefit, the baseline technique is unexpectedly competitive with current state-of-the-art meta-learning algorithms. It is the goal to encourage future advancement in the field by making the source code and model implementations freely available.

1.4 Research Methodology

We have adopted the technique of few-shot learning to get the good results from small datasets. We will be modified and extract small datasets of images from public datasets. We gathered four datasets and extracted 20 of the best samples from each. First, we employ a convolutional neural network strategy to derive accuracy from the Few-shot learning technique. Following that, we used a reptile technique based on Meta learning. This technique gave positive outcomes for us. We also tweaked the reptile approach to improve the needed accuracy by increasing the depth and modifying the learning module at the back propagation.

1.5 Summary

After applying all these methodologies, the extracted results were satisfactory and comparable with machine learning techniques where large datasets are required for training. Due to these points, we can clearly say that few shot learning has become the need of the hour and our accuracy in this paper will clearly show that we can get the better accuracy from the small datasets also if the right algorithm and approach will be used with few shot-learning. A lot of researchers have started to include few shot learning techniques into their research areas because it’s helping the researchers to get the required results in shorter interval of time with less computational cost.

CHAPTER 2

LITERATURE REVIEW

Algorithms that use a minimal number of labeled instances are known as "few-shot learning algorithms." There has been a great deal of work put into finding a solution to the data efficiency problem. Researchers have derived a lot of new algorithms by using the few shot learning techniques and dataset distillation is the clear example of it. Dataset distillation is the process of extracting the best examples from the set of data which means converting a large data (with noisy or random data) to a small dataset (having best examples). Distillation is a advances area of neural network, which mainly focus on the giving the better accuracy by extracting the knowledge from a large pre-trained model and transferring it into a smaller model [11].

There is evidence that Dataset Distillation back propagation may be used to produce tiny synthetic datasets that can train neural networks to roughly the same accuracy as when training on the original datasets [10]. To achieve their goal, first they have derived the network weights as a differentiable function for the training data. To optimize the model, they didn’t optimize the network weights instead of that they optimized the images. They first compressed the dataset of total 60,000 images to just 10 best images then optimized those 10 example images to get better accuracy. After training on just one such distilled picture per class, networks may achieve over 90% accuracy on MNIST. As a result of Dataset Distillation, it took hours of training to create distilled pictures. Problems with relatively few samples per class do not lend themselves to this technique. They also tried the same technique on some other dataset just to verify the credibility of the algorithm and got good results as compared to the other alternative techniques which were already applied on those datasets.

In another study [12], researchers have used the same technique of dataset distillation to extract the better accuracy from the smaller dataset, but they also sued few labels for the extracted smaller dataset and called this technique as labelled distillation. By using the labels for the smaller extracted dataset, they got the better accuracy then already available large dataset where labeling was used.

In a survey [13], researchers have shared their findings about the process of few-shot learning in which they shared the same reasons why larger datasets have become difficult to handle and why people should shift towards the smaller datasets and adopt the few shot learning techniques. As per the researchers, few shot learning technique has the capability to generalize the provided new tasks which will be containing just few samples with supervised or labeled information. They conducted a thorough survey to fully understand the working of the few shots learning technique. They compared the few shots learning technique with few existing machine learning techniques to find out the difference and flaws in few-shot learning technique. Based on this experiment, they divided the working of few shots learning into three different categories and performed their research as per those categories. Moreover, they found out that in case of few shots learning, empirical risk minimized in not reliable.

The usage of few-shot learning has started to show its impact in a lot of domains. Most of the researchers are trying to train the models with the help of this technique as training with large datasets has become difficult. But it is also important to generalize the dataset in a best way so that better results can be extracted during the few-shot learning technique. (Ravi & Larochelle, 2016) have discussed the best techniques to generalize the dataset to make it useful for few shot learning techniques.

In another study, researcher has suggested a Less than One-Shot (LO-shot) learning [14]. In this technique, the model must be trained to recognize N classes using just M training samples when M<N which means he provided less classes for the training but use more classes then for

prediction. To achieve this high level of data efficiency, the training samples are given richer labels he named them as soft labels. This technique is more related to soft labels. For reference, they have quoted an example of an alien who came on earth for the first time to find the unicorn with no real picture or recognition. People on Earth show him two photographs, one of a horse and the other of a rhinoceros and tell him that a unicorn is a hybrid of the two animals. Alien would be able to recognize three different creatures with this knowledge. This approach has helped in defining the one-shot learning in the real environment and the more dependency lies on the soft labels to extract the more information. In this paper, researchers have especially aimed to research the theoretical foundation of LO Shot learning so that they can find a better and easier model then deep neural network. He showed analytically that it is feasible to create soft-label prototypes that can be fitted using a KNN classifier. Soft labeling needed a great deal of human expertise about the dataset.

CHAPTER 3

Methodology

3.1 Dataset Selection

The need of the dataset is very important for the machine learning, as without it our model would be unable to learn any new information [17]. Basically, it prepares the AI model to develop and build the understanding to acknowledge and recognize any information. For the purpose of our research, we selected four public Dataset which description are given below in the table.

Name	Total Classes	Training Examples	Test Examples	Accuracy
MNIST	10	50,000	10,000	99.9 %
Fashion MNIST	10	50,000	10,000	99.9 %
cifar10	10	50,000	10,000	99.7 %
PCB	8	13,500	2,700	98.2%

3.1.1 MNIST

The MNIST dataset is from the National Institute of Standards and Technology (NIST). They collected the training dataset comprises of handwritten digits from approximately 250 different people, and out of those 250, half of them are high school students and the remaining 50% are from the Census Bureau. The handwritten digital data have the got the same proportion of test set. In MNIST dataset there are total of 60,000 images in the training dataset and of about 10,000 patterns in the testing set, where each one of those images are about 2828 pixels size and with 256 gray levels [8]. It is very simple to download the dataset online and as some of the examples from the MNIST corpus are shown in Figure below.

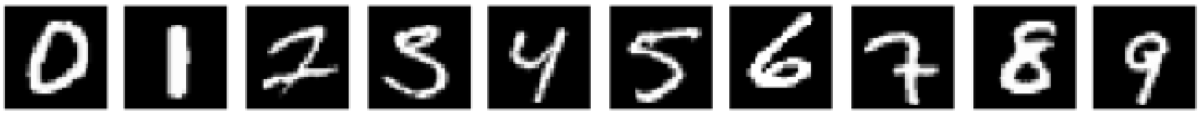


Figure 1 MNIST dataset images

3.1.2 Fashion MNIST

Fashion MNIST contains 70 thousand images. Each of the 70k photos in Fashion MNIST has a spatial dimension of 28 x 28. All the photos are grayscale and have a single channel. The total number of classifications is ten, with each category including seven thousand photos. The training set has 60, 000 samples, whereas the test set contains 10,000. Fashion-MNIST is designed to be a drop-in replacement for the original MNIST dataset for benchmarking machine learning methods, as it has the same image size, data format, and training and testing split structure as the original MNIST dataset.

Fashion-MNIST is based on the assortment on Zalando's website2. Every fashion product on Zalando comes with a series of photos taken by professional photographers that showcase various features of the product, such as front and back views, details, and looks with a model and in an outfit. The original picture has a light-gray background (hexadecimal color: #fdfdfd) and stored in 762 × 1000 JPEG format. For efficiently serving different frontend components, the original picture is resampled with multiple resolutions, e.g., large, medium, small, thumbnail, and tiny.

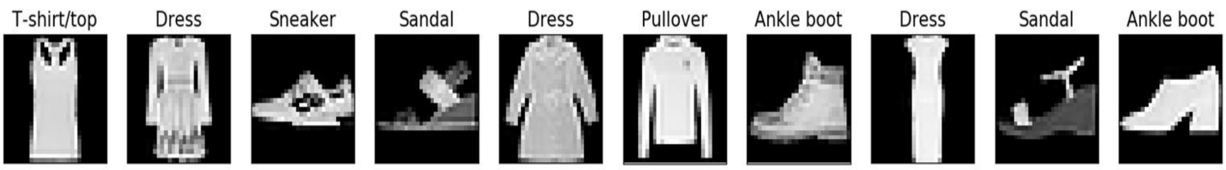


Figure 2: Fashion MNIST dataset images

3.1.3 Cifar-10

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

Figure 3: Cifar-10 Dataset Images and class name

3.1.4 PCB

PCB dataset is the data of the red blood cells of human body. For the training purposes of few shots machine learning we took the blood cell dataset as there are eight different types of blood cells: monocytes, lymphocytes, neutrophils, eosinophils, basophils, macrophages, erythroblast, and platelet. This is color dataset with 13500 training images and 2700 test images sample, each of that image of size (160, 160).

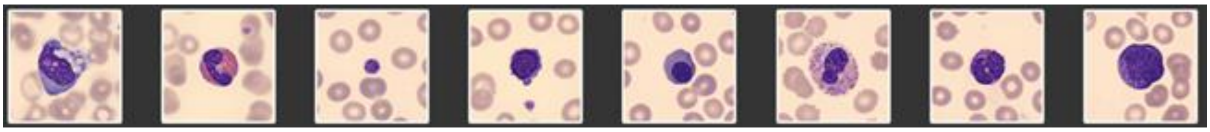


Figure 4: PCB blood cell Dataset images

3.2 Sample Selection for Few-Shot Learning

For the purpose of few shots experiment on the selected we have selected only 20 examples for each class from the dataset, which contains only 200 examples for practically all 10 classes (excluding PCB, which has 160 samples and is also 20 examples per class). Which are only 0.4 percent of training examples from original. It is very common in metric training networks to have a large testing dataset in order to get a fair numerical representation of the accuracy of model on a dataset of high variance. 0.4% means model has seen a very small variation in the dataset and thus has learned exceptionally good features representable enough to distinguish similarity from dissimilarity.

Name	Total Classes	Training Examples	Test Examples
MNIST	10	200	10,000
Fashion MNIST	10	200	10,000
cifar10	10	200	10,000
PCB	8	160	2,700

3.3 Objective Function

After selecting few examples from dataset. We used four different approaches to train the model and compared the outcomes. These approaches are following:

- CNN Network
- Siamese Network with Triplet loss
- Reptile – Meta Learning
- Our approach name Stomatopods Algorithm inspired from reptile.

3.3.1 Convolutional Neural Network (CNNs)

Deep Learning has become an important area of research in the Machine Learning community. In addition, it has gradually become the most widely used computational approach in Machine Learning, achieving excellent results in several complex cognitive tasks that match or exceed what is offered by human performance. One of the advantages of Deep Learning is that it can learn large amounts of data as shown in

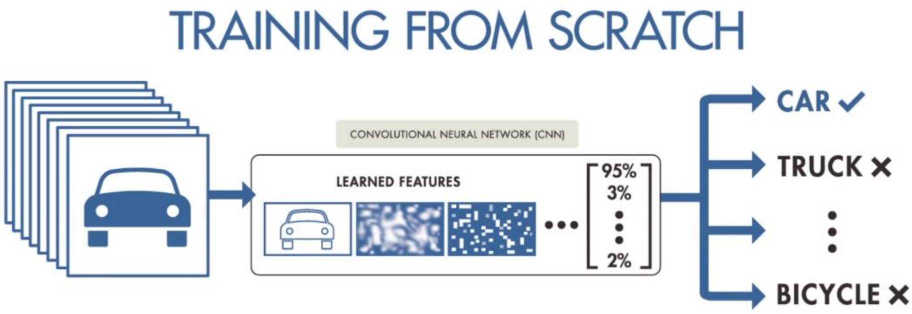


Figure 5 CNNs Feature Learner cycle

The field of Deep Learning has grown rapidly in recent years and is widely used to successfully tackle a variety of traditional applications. Most importantly, Deep Learning outperforms known machine learning technologies in many areas, including cybersecurity, natural language processing, bioinformatics, robotics, and medical information processing and control. Recently, machine learning (ML) has been widely used in search and has been incorporated into various applications such as text mining, spam detection, video recommendations, image classification, and multimedia concept search. In the area of Machine Learning is currently one of the most important research trends due to its remarkable success. Convolutional neural networks (CNNs) are one of the most common and widely used DL networks. Thanks to CNN, DL is now very popular. The main advantage of CNN over previous ML based methods is that it automatically detects important features without human supervision and is the most used. That's a reason why computer vision delved into CNN by introducing its key feature extraction methods. CNNs are widely used in classification, object detection, object recognition, representation learning and few shot learning methods, even if they are not used directly, they take a huge share in feature extraction backbone networks.

Mobile Net Architecture are a type of lightweight deep convolutional neural network that is much smaller and performs far better than many other popular models. They are also low-latency, and low-power Convolutional neural networks which can easily train on the local machine Because of their compact size, this deep learning models are ideal for application on mobile devices. Mobile net has been chosen for experimentation in this study, the architecture and results are presented below.

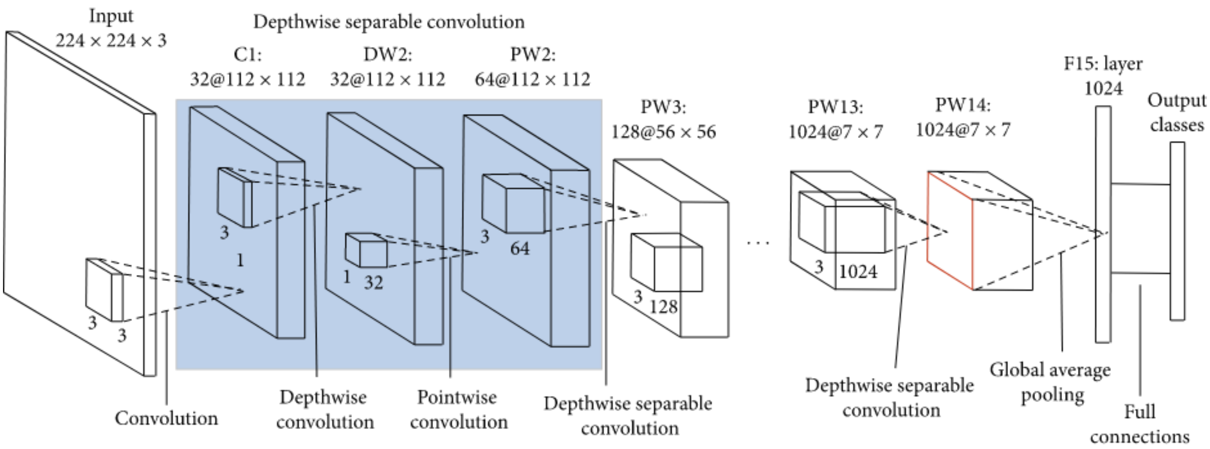


Figure 6: Mobile Net Architecture

3.3.2 CNN Training Hyperparameters

A Convolutional neural network is typically trained in two stages: The input is totally transferred across the network during the forward phase. Gradients are backpropagated and weights are updated in the backward phase. The hyper parameter selected for the training this architecture is given below in the table.

Hyper parameter	**Image Size**	**Training Example**	**Batch size**	**Learning rate**	**Epoch**
Value	(160, 160)	20	8	0.00001	50

3.3.3 CNN Accuracy Plots

We can observe from the plots given below that the model overfits on the training sample and performs poorly on the validation dataset. Which clearly indicates that It will also perform poor on the test set.

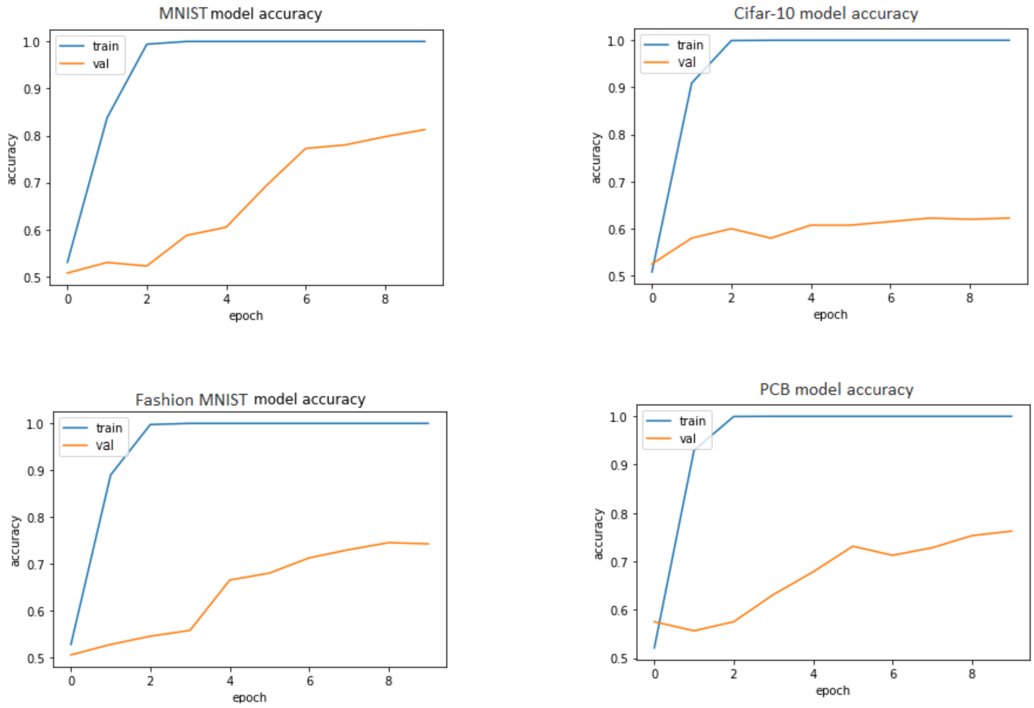


Figure 7 CNN Training Accuracy plots

3.3.4 Results

Model Accuracy is calculated on complete test set of 10,000 training example for first three datasets which are MNIST, Fashion MNIST and cifar-10 and 2700 on PCB test examples. Below is the bar chart of dataset accuracies.

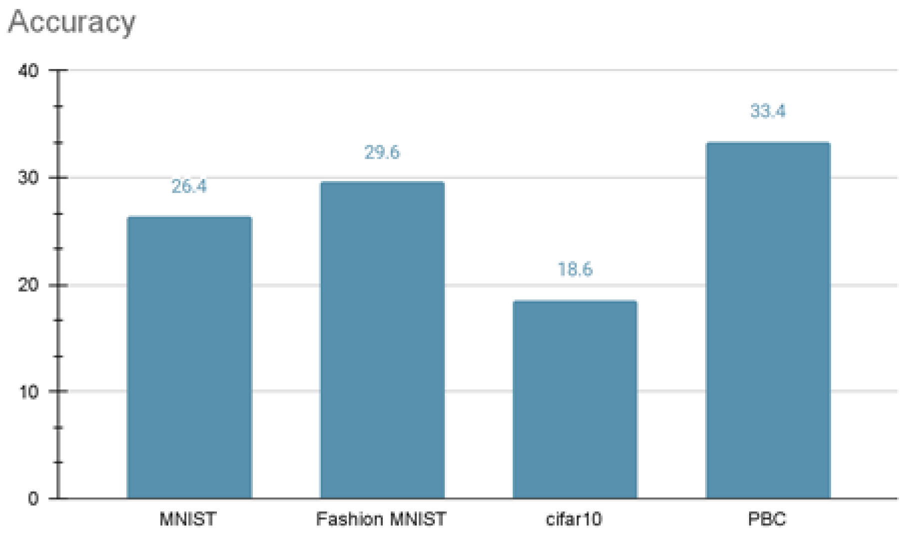


Figure 8 Accuracy Bar CNNs

3.4 Siamese Networks

Neural networks are practically perfect at almost every activity in the present Deep learning era, but they rely on additional data to perform properly. However, we can't always rely on more data for specific problems, such as recognition-based tasks, to address these issues, researcher have developed a new type of neural network design known as Siamese Networks. Siamese network is a type of CNN in which the primary objective is not to learn the classification boundary between classes but to learn similarity and dissimilarity. They are widely used for recognition-based tasks where they are exceptionally good at performing single shot learning.

Siamese work like taking two different input samples which are passed via same networks with same architecture, same weights, and hyper-parameters. Each of the two networks output an encoding which may be called an embedding. Embeddings are than compared and Siamese network must decide which of the pairs is similar or dissimilar hence learning not the classification, but the representation of the input pair provided which helps in learning the similarity or dissimilarity instead of learning the decision boundary. Loss functions used include the binary cross entropy loss, contrastive loss, triplet loss or their variants. Siamese networks learnt discriminative features to learnt to recognize the unfamiliar samples from unknown distribution. Hence it learns to learn the similarity and dissimilarity between images given the images be sampled from familiar distribution, which means such a network if trained on face images, will perform recognition on unknown images but if the input images are of vehicles, then its performance will not be good because it had learnt the feature representation for face features.

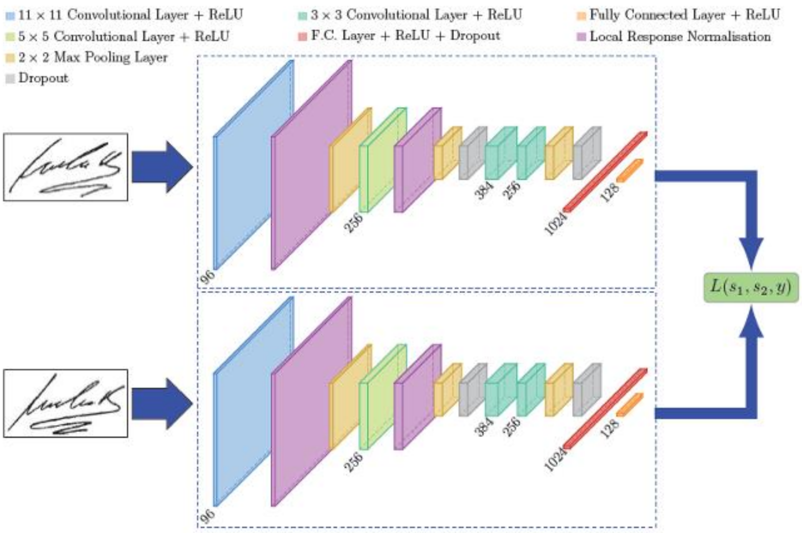


Figure 9 Siamese Network

Loss functions used for Siamese networks are:

3.4.1 Contrastive Loss:

For contrastive loss, two inputs need to be passed through the network, these inputs can be similar or dissimilar and this is reflected by the label Y which if 0 denotes those images are dissimilar and if 1, denotes similarity.

$$loss(distance, Y)$$
$$= \frac{1}{2} \times Y \times distance^2 + (1 - Y) \times \frac{1}{2} \times \max(0, margin - distance)^2$$

3.4.2 Triplet Loss

Triplet loss as the name implies, it takes three input images at each iteration which are referred to as anchor sample, negative sample, and positive sample.

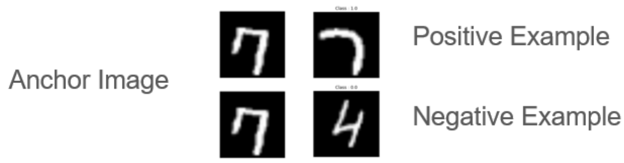


Figure 10: Triplet Example

Anchor sample is chosen which stands as a reference to the comparison and the goal is to maximize the distance between it and negative images and minimize the distance between it and positive images:

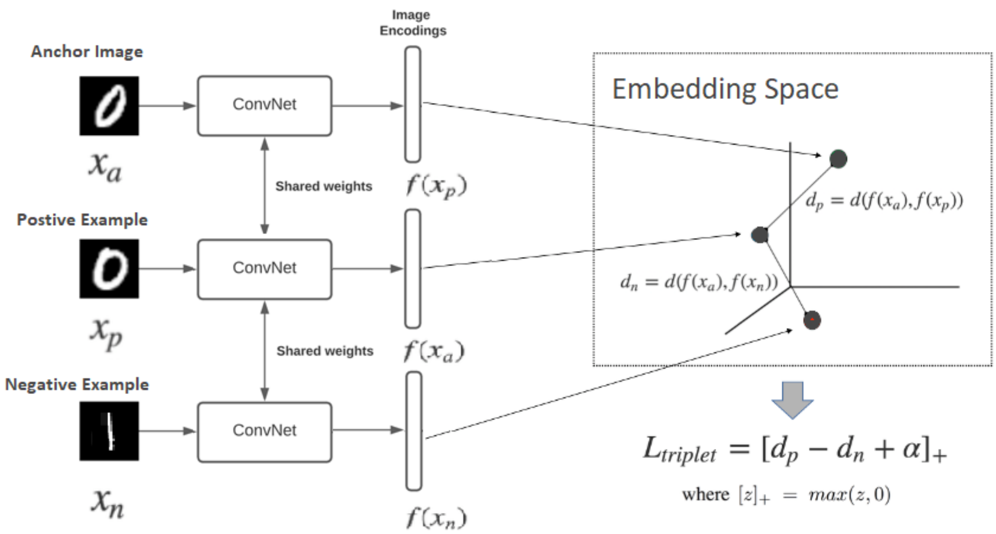


Figure 11: Triplet Network with Triplet Loss

- Take an anchor object which needs to be chosen as point of comparison for other two samples.

- Sample a positive image which belong to the anchor image.
- Sample a negative image which is dissimilar to the anchor image.

Loss is calculated as:

$$loss(d_1, d_2) = \max(d_1^2 - d_2^2 + margin, 0)$$

3.5 Triplet Network

Triplet Network is the second approach we selected for few shots learning. The triplet network is a step forward from the Siamese network. To avoid overfitting, positive and negative distances are learned simultaneously by using this combination. Feature extractor is a CNN based model used for getting the embedding of triplets during the training and evaluation process for the purpose of calculating the triplet loss. The architecture of the Feature extractor we selected is given below.

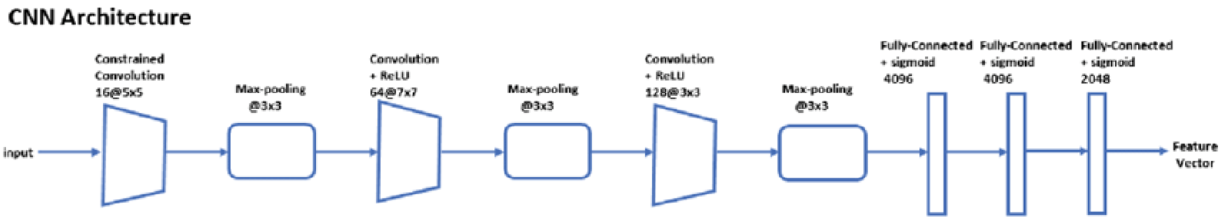


Figure 12: Feature Extractor architecture

3.5.1 Triplet Training

The hyper parameter selected for the training this architecture along with the training accuracy graph is given below in the table.

Hyper parameter	**Image Size**	**Training Example**	**Batch size**	**Random Samples**	**Epoch**
Value	(200, 200)	20	4	10	10

Training Graphs

These are the training graphs of Triplet Network using triplet loss of all four datasets. The validation split during training for this network was set to 0.2.

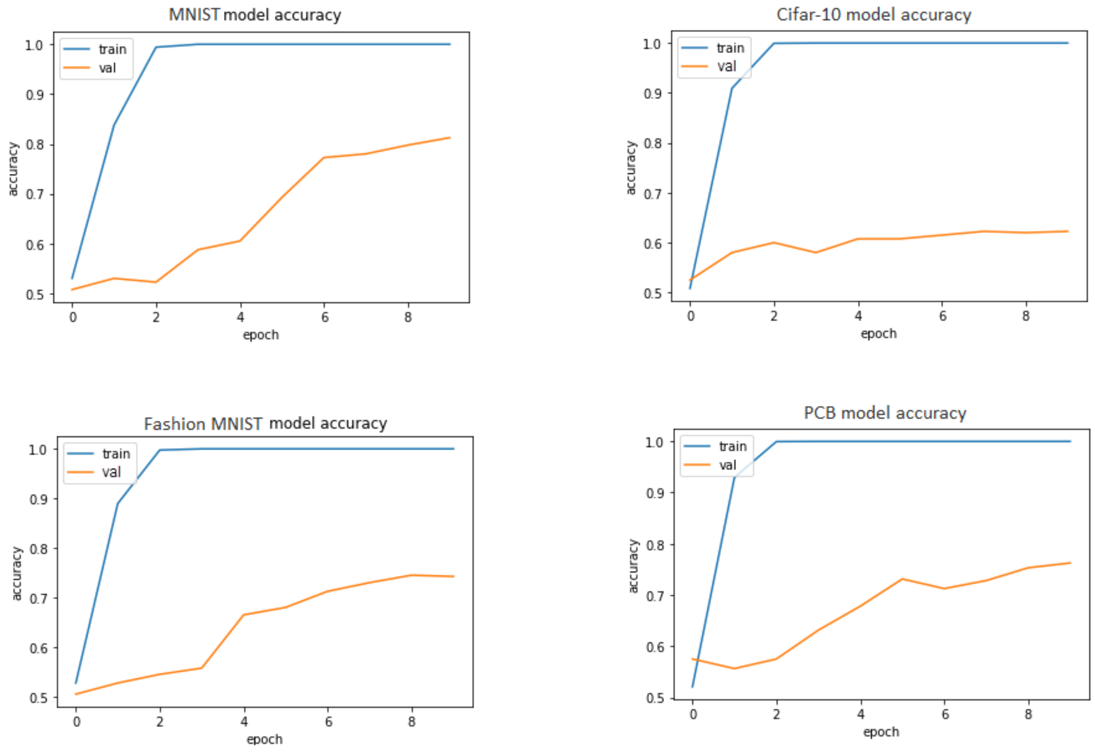


Figure 13: Triplet Network training Accuracies

3.5.2 Triplet Network Results

While training it is observed that such networks are performing good on validation set but when tested on 10,000 test examples perform poor. The result we obtain using this triplet network technique with comparison of previous CNN technique are given below:

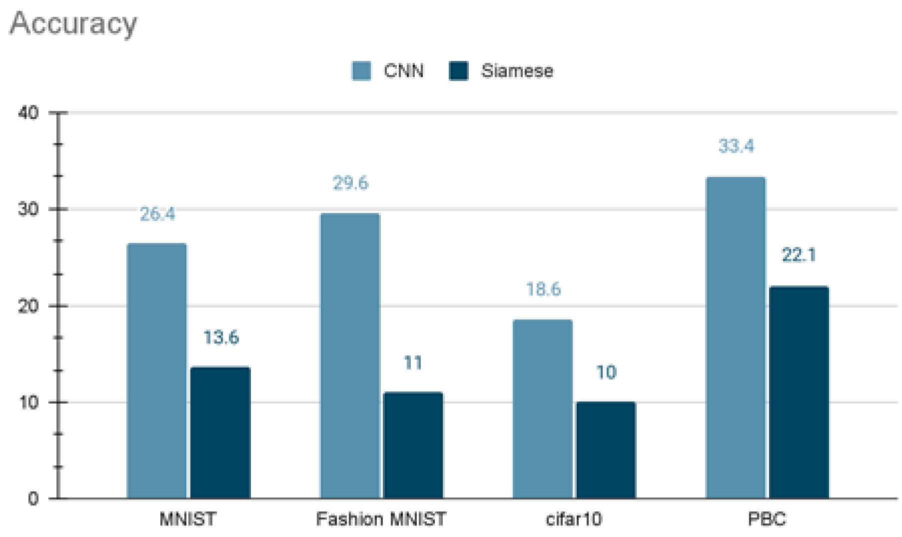


Figure 14: Accuracy Bar of CNN with Siamese

From this graph of accuracies, we can interpret that Siamese Network with triplet loss are not performing better result as compared to CNN technique.

3.6 Reptile-Meta learning

Meta learning can be generalized to different situations by dividing a particular task into two functions. The first feature usually provides quick feedback on a particular task, and the second involves extracting information learned from previous activities. It resembles human behavior and usually derives knowledge from unrelated activities and previous experience. There are three common approaches generally adopted for meta-learning:

- Metric-based: Learn an efficient distance metric
- Model-based: Use Recurrent network with external or internal memory
- Optimization-based: Optimize the model parameters explicitly for fast learning

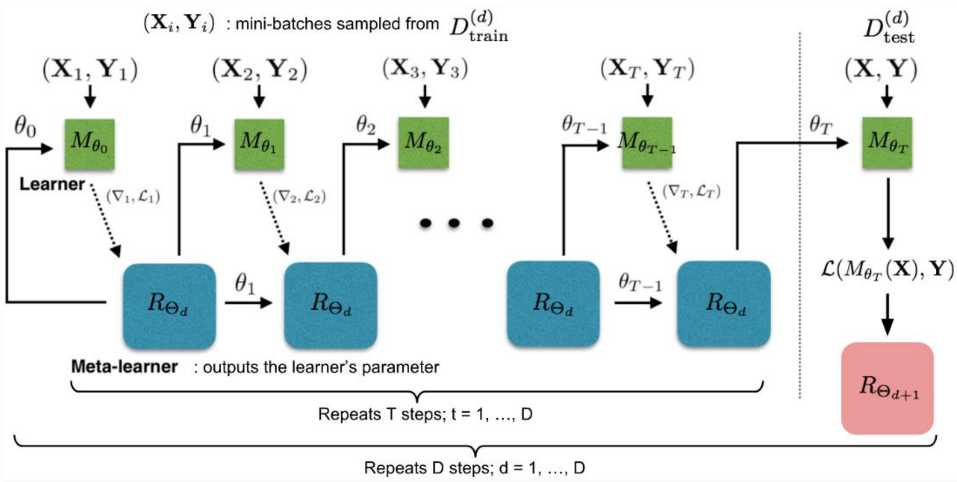
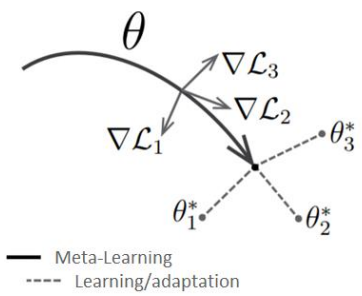


Figure 15 Meta-Architecture Model-based Recurrent network

Reptile uses a Random Gradient (SGD) to initialize model parameters instead of performing various resource-intensive calculations. Reptile can be regarded as an application of the shortest descent algorithm to the meta-learning setting and is similar to the first-order MAML that only needs black-box access to an optimizer such as SGD or Adam, with equal computational efficiency and performance.



Moreover, experts believe the Reptile algorithm is not necessarily superior to MAML in terms of learning performance, but it's relatively simple to implement since it relies on SGD.

3.6.1 Reptile (Training)

From the graph we can see that model learn better than all previous approaches.

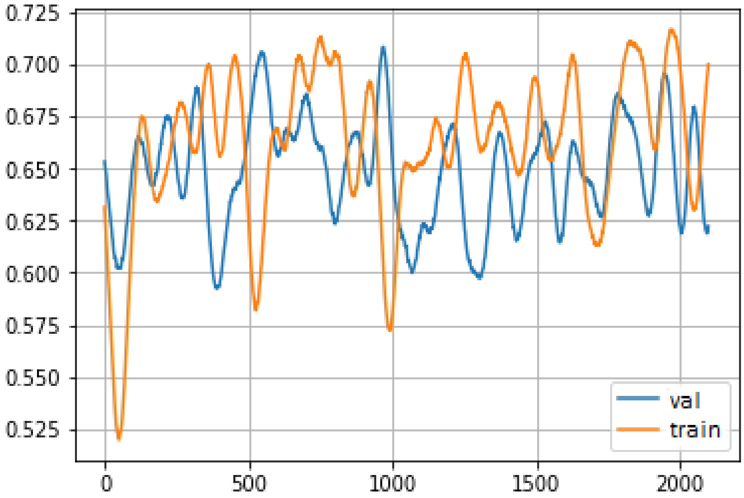


Figure 16: Reptile training graph on MNIST dataset

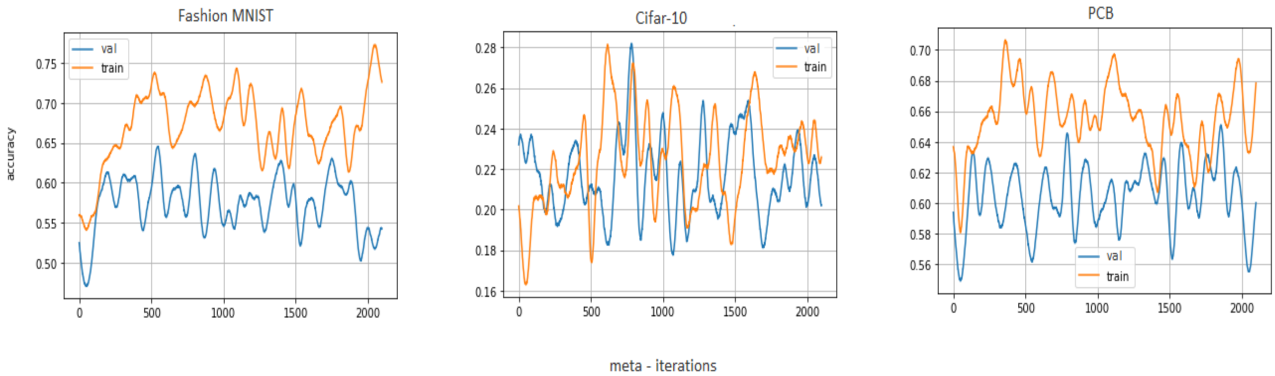


Figure 17 Reptile Training graphs

3.6.2 Results

Here this is the again comparison plot between Reptile and previous approaches. From the plot we can see that Reptile perform much better than previous approaches.

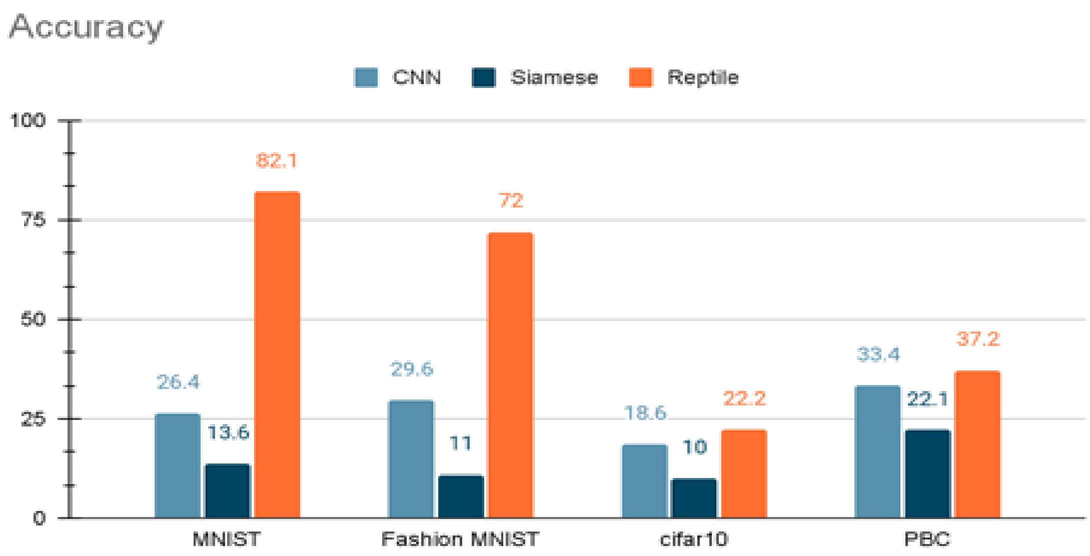


Figure 18 Retile comparison accuracy bar chart

3.7 Our Approach

Using the meta learning approach, I was able to get an outstanding outcome. In order to achieve better outcomes, we should strive to investigate this approach and tweak this sort of network to achieve more positive outcomes. We proposed a Meta Learning algorithm named as Stomatopods which has more architectural depth as compared to reptile architecture with a different weights update policy during training as compared to conventional training and with different data feeding techniques to network.

3.7.1 Mini Batches and Data Loader

Making of mini batches of dataset in such a way that data loader assign different number of label index to each class in a minibatch is essential for training this model because we don't force network to predict the same class on the same index in the last layer during training as in conventional CNN networks. We have 200 training images in our case of MNIST, Fashion MNIST and Cifar10 dataset. For the achievement of our goal algorithm necessitate the use of mini batches of these datasets. The following figure shows the process of making mini batches:

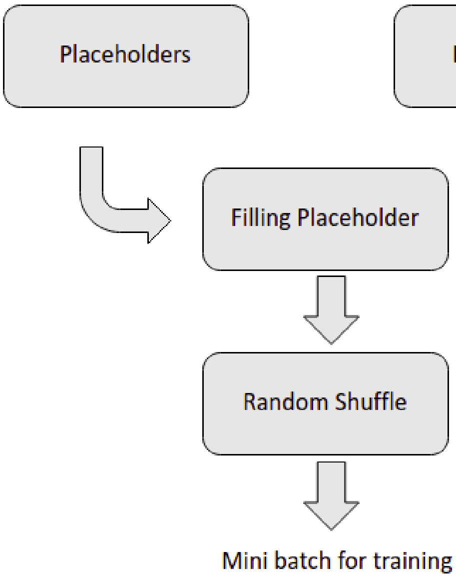


Figure 19 Mini Batches flowchart

First and foremost, a placeholder for images and labels is constructed. These are the empty lists where the images and labels will be collected. We set the 5 shoots for training, and we have 10 classes. so, the placeholder of labels has a shape of 10x5 and placeholder for image have a shape of *10x5x28x28xd*. Where *d* is the depth of image. In the case of gray scale image *d* is *1* and in the case of RGB image *d* is equal to *3*. We randomly sample the label subset. It contains all the possible classes but not in ordered form. Let's take an example of MNIST in which the first label is zero and it is always placed at the index of zero for the training of conventional CNN networks. In our case we don't force to network to predict the zero at zero index. We allow the network to predict zero at another index like 4. During training we don't define the index for each class in dataset. It is selected randomly. Once we have a placeholder and randomly sampled labels then we will fill the placeholder according to labels. After that we shuffle the images and labels to make mini batch for training of network.

3.7.2 Dataset Post processing and Implementation in TFRecord

As previously mentioned, we used four separate datasets. These datasets arrived in a variety of formats for the training purpose. MNIST, Fashion MNIST and cifar10 are available in TFRecord formats and The PCB dataset arrived as a raw picture, which was placed in a directory. TFRecord is TensorFlow’s binary storage format. When we have a significant amount of data, it comes in handy for handling TFRecord. We converted the PCB dataset to TFRecord and used it for the training of our Stomatopods Algorithm. Below figure shows the data loading pipeline.

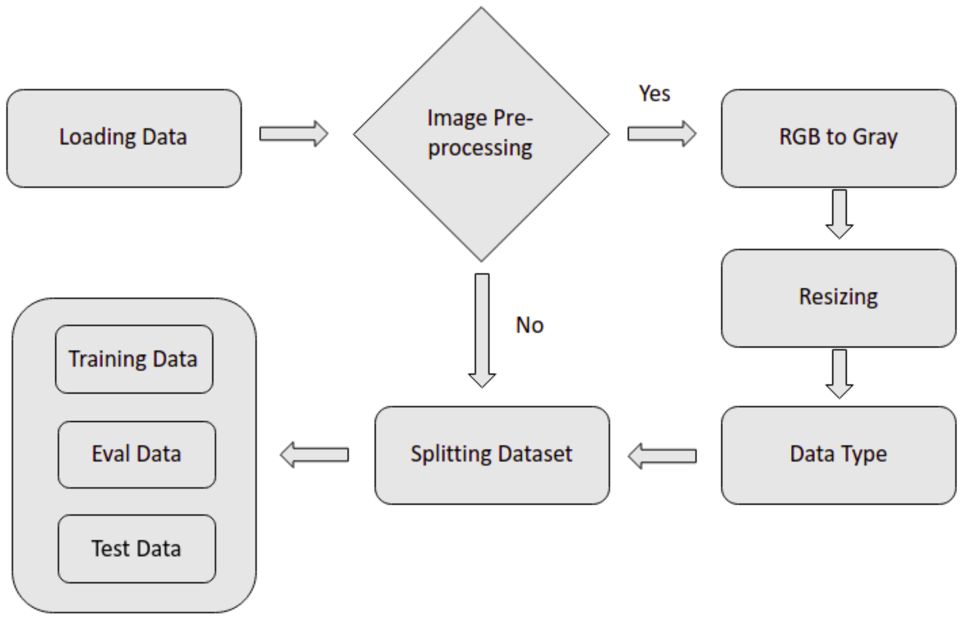


Figure 20: Our approach Data loading pipeline

The data is pre-processed after it is loaded from disk. It is entirely up to us whether to process the data. The disadvantage of not performing pre-processing is that all images must be scaled before being delivered to the network during training. If we don't resize the image before training, it will start resizing during training, which will slow down the process as a result. It is a nice choice to complete all necessary processing prior to training. The initial step in the pre-processing procedure was to convert RGB photos to grayscale images. Only the MNIST and Fashion MNIST datasets are subjected to this pre-processing. The datasets are already grayscale, still required eliminate of channel to make them grayscale in TensorFlow because it read images as in three channels by default. We used bilinear interpolation to resize the images and resize the photos to *(28 x 28)* image size after the conversion. We also modified the data type after it. The image's default data type is an *8-bit* unsigned integer. This data type is not suited for neural network training. The float data type with *32 bits* is suggested for the TensorFlow library. As a result, we updated the image data type to *float32*. After the data has been pre-processed, the dataset is divided into three sections: training, assessment, and testing. The division of the dataset is given in the table below:

Set	Portions
Training	60 %
Evaluation	20 %
Test	20 %

3.7.3 Stomatopods Model Definition:

To increase performance, we tweaked the original reptile network and named Stomatopods. The following diagram represents the entire network:

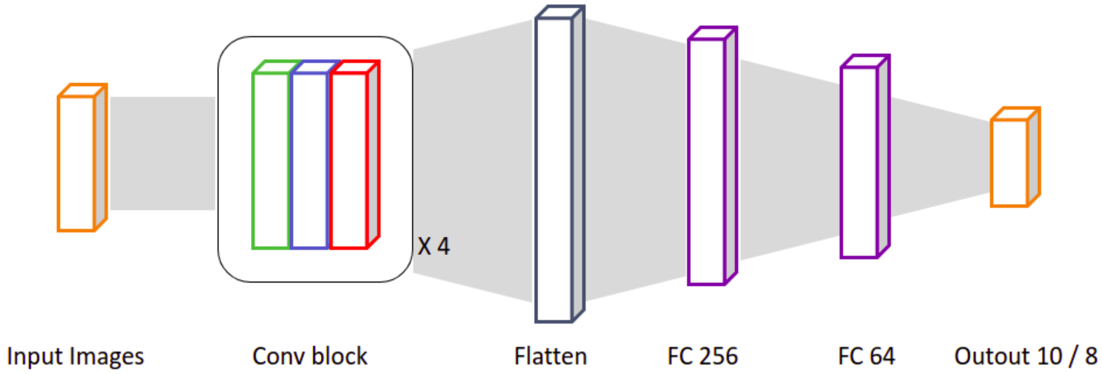


Figure 21: Stomatopods Model

The input layer is the initial layer of the network. It has a shape of *bx28x28xd*. Where *b* is the mini batch size and *d* is the depth of image. The image in the input batch is passed through a sequence of Conv blocks. The Conv blocks are the combination of convolution layer (shown in green), batch normalization layer (shown in blue) and ReLu activation layer (shown in red). We used *64* filters of size *3 x 3* with stride of *2* in each Conv block. There are four different Conv blocks in use, which are connected in a series. The output last Conv block is flattened by using Flatten layer implement in TensorFlow. This layer has no parameters but is useful for converting three-dimensional feature maps to two-dimensional feature maps. After that, two different totally connected layers are placed on top of each other. The first layer is FC 256 with *256* neurons with ReLu activation. The second layer is FC 64. The only difference between it and FC 256 is that it has 64 neurons. The output layer is the network's final layer. This layer has 10 or 8 neurons Depending on the number of classes. There are 10 neurons in the MNIST, Fashion MNIST, and CIFAR 10 datasets, and 8 neurons in the PCB dataset. SoftMax is the layer's activation function. On the following page, there is a diagram with a complete description of the model, including input and output shapes.

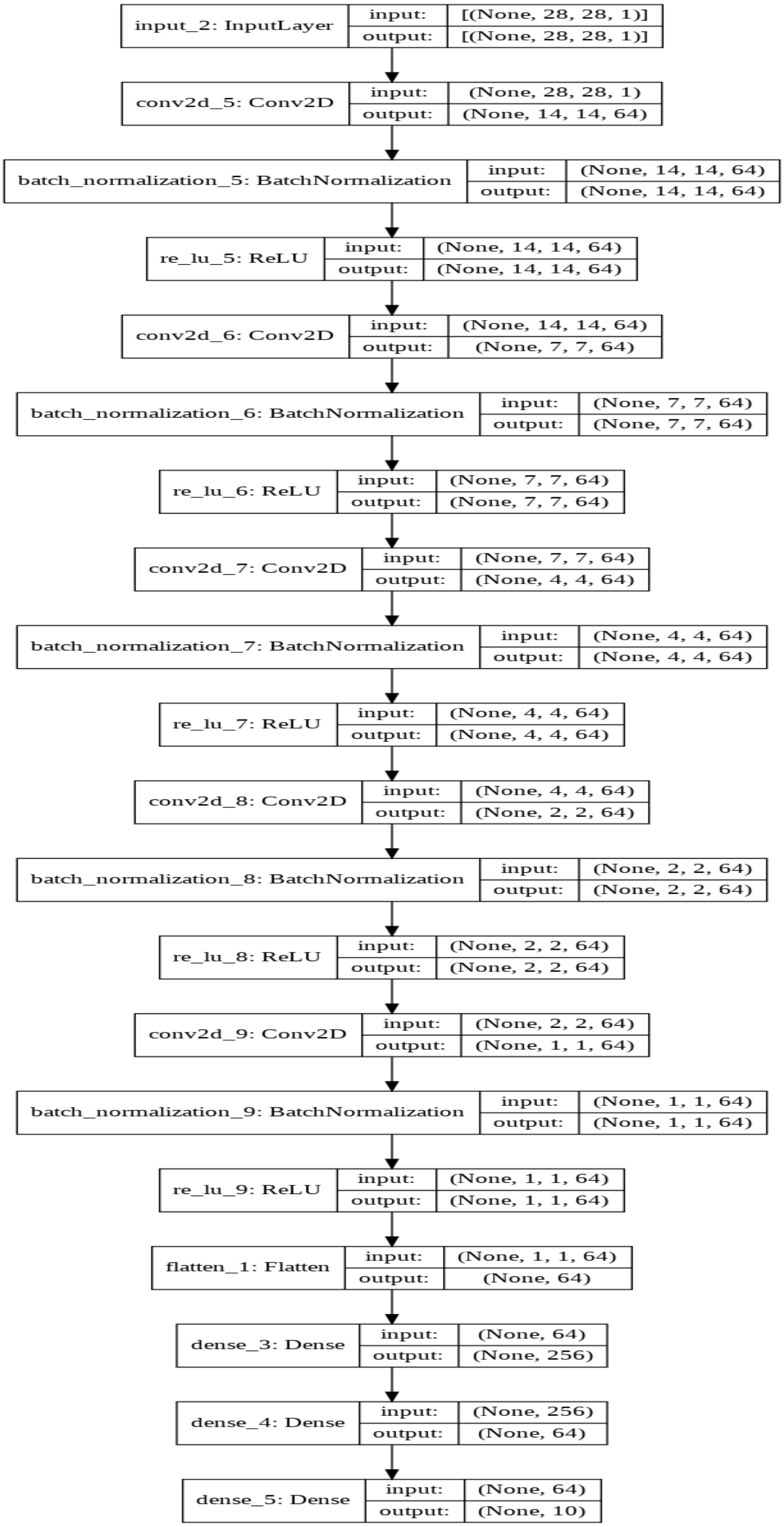


Figure 22 Stomatopods Architecture details

3.7.4 Model Training

The training of the Stomatopods model differs from that of the CNN networks. The difference is in the weights policy update. Let's specify several basic parameters and abbreviations before getting into the weight updating mechanism:

Abbreviation	**Description**
E	Total number of Meta iterations.
e	Current Meta iteration
ms	Meta step size define as 0.25.
$ms(e)$	Current meta step size.
W	Weights of the model
Wo	Old weights of model before training.
Wn	New weights of model after training.
W^*	Optimal weight of model.

The total learning cycles are referred to as total meta iterations in meta learning, while the present training iteration is referred to as meta iteration. The model weights are modified after each iteration. Consider the following diagram, which depicts one Stomatopods algorithm learning cycle:

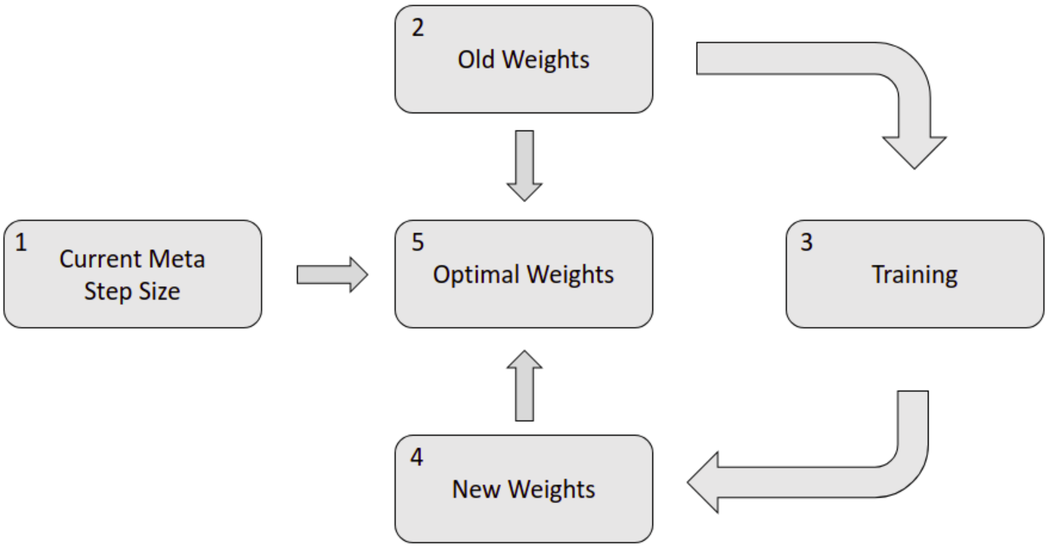


Figure 23 Stomatopod's algorithm learning cycle

To begin, the current meta step size is computed using the equation below.

$$ms(e) = \left(1 - \frac{e}{E}\right) * \ ms$$

Following that, the model's old weights are preserved for subsequent use, and the model is trained on a tiny batch obtained from the data generator. During training, a small batch is used to calculate loss, and weights are updated using a gradient descent optimizer. We get our new weights after the training. These weights are not the model's final weights. Using the current meta step size, old weight, and new weights, we determine the ideal weights. The following equation is used to calculate the best weights:

$$W^* = Wo + (\, Wn - Wo) * ms(e)$$

It will replace the existing weights once the optimal weights are calculated. This cycle is repeated until the total meta iterations are reached.

3.7.5 Pseudocode for updating parameter

Initialize α, the initial parameter vector

for step(iteration) 1,2, 3… **do**

Sample a task randomly *R*

Perform $i > 1$ steps of Stochastic gradient descent (STD) on task *R*, starting with parameters α, ending with parameters *W*

Update: $\alpha \leftarrow \alpha + ((W - \alpha) * ms(\epsilon))$

end for
Return α

3.7.6 Training of Stomatopods

Here are the Stomatopods training accuracy plot trained on 2000 meta iterations. This plot shows that stomatopods learn better.

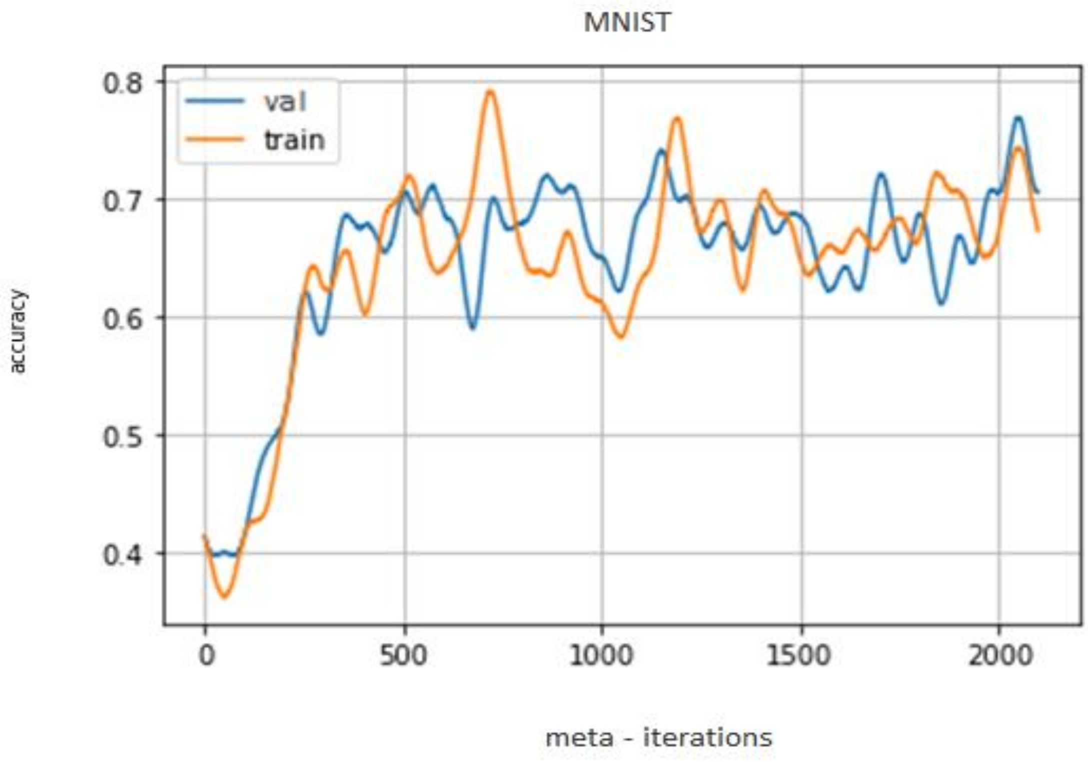


Figure 24 Stomatopods MNIST training accuracy plot

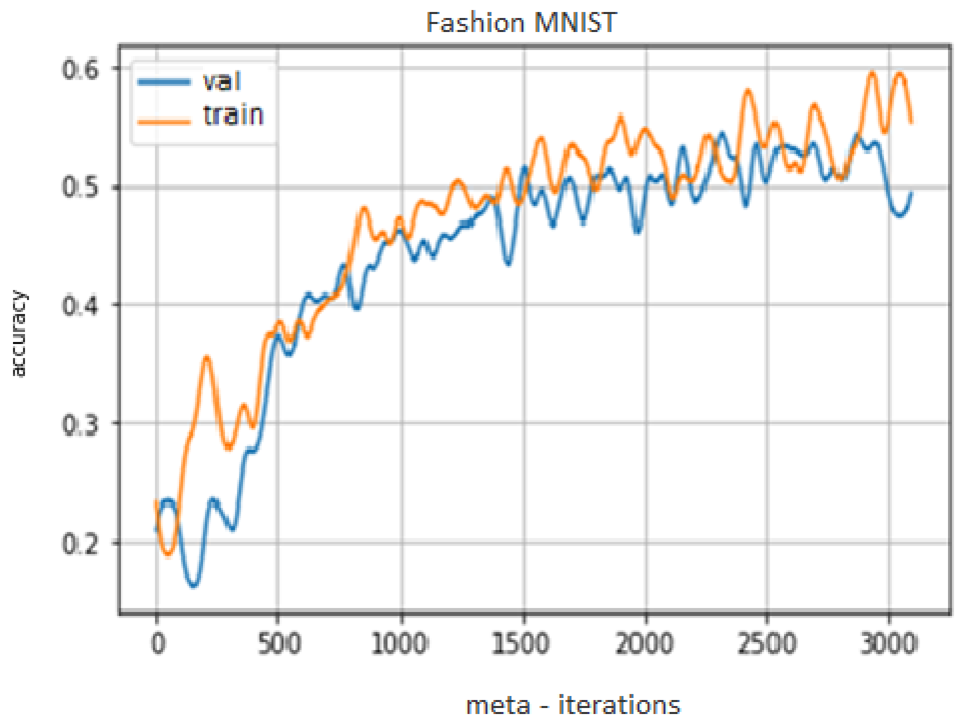


Figure 25 Stomatopods Fashion MNIST training accuracy plot

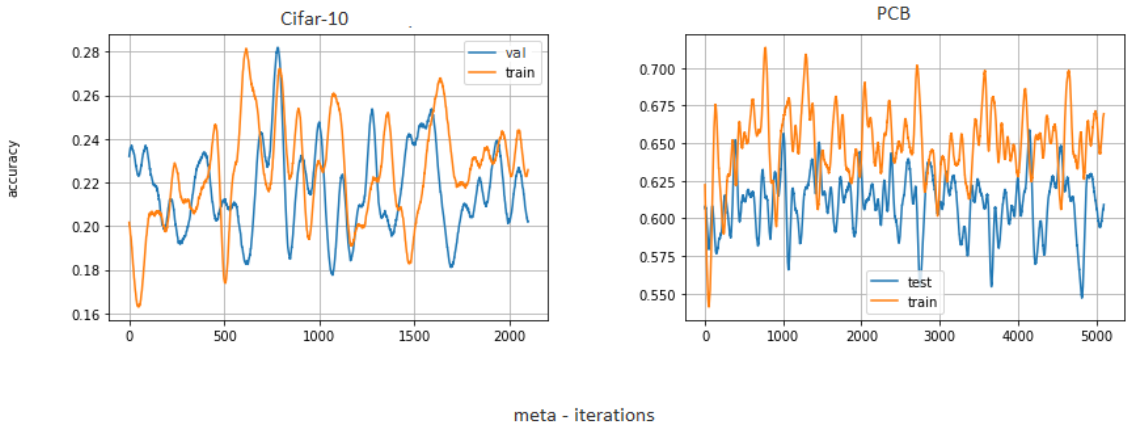


Figure 26 Stomatopod's training accuracy plot

CHAPTER 4

Results and Discussion

For few-shot classification tasks, we test our algorithms on four common datasets and make a decision based on roughly ten thousand test samples. The accuracy of all four techniques is plotted here. Our method outperforms the competition on every dataset. On MNIST FASHION MNIST, CIFAR 10 and PCB, we achieved 86.9%, 75.3, 35.7, and 63.9 percent accuracy, respectively, using our approach.

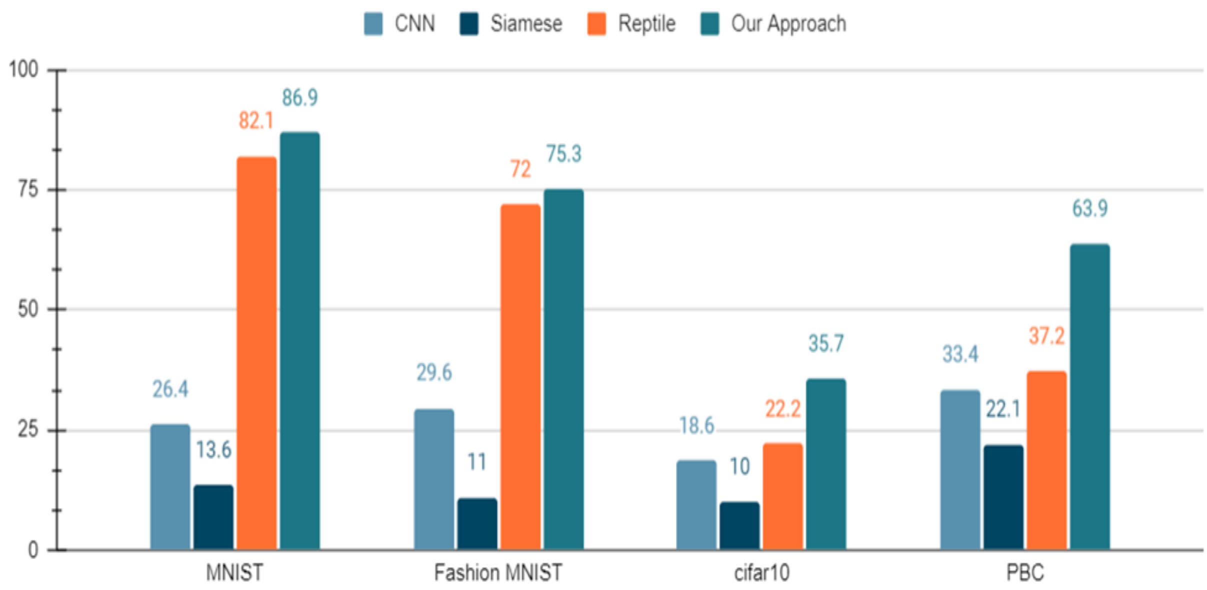


Figure 27 Our Approach Comparison Accuracy bar chart

The key reason for our approach's impressive results is that it uses a different learning strategy than conventional learning and uses a different data feeding technique. It is also observed that Normally, CNN performs well with large datasets, but it is difficult to achieve fantastic results with only a few examples. The Meta learning approach, on the other hand, is excellent and provides greater results with only a few shots of learning.

4.1 Intra-Class and inter Class variance:

The variation between multiple observations of an individual (intra-class variance) and the variation between subjects (inter-class variance) define the performance of any algorithm. Within class (intra-class) variance is important to control separation via attracting elements of the same class. The intra-class covariance matrix can be used to evaluate covariances of within-class embedding errors and then visualize them via initial PCA component. Siamese network got popularity from face net in which it learned cosine similarities of different faces and is a good approach to find the similarities index within same class. Face net is a recognition algorithm and objective of this model is to find either the face is similar or not. It is observed for such type of objective Siamese network performed better. Reducing the distance between the examples and increasing it between the examples and center point (zero vector, local mean, or global mean) leads to low intra-class variation and high inter-class variation.

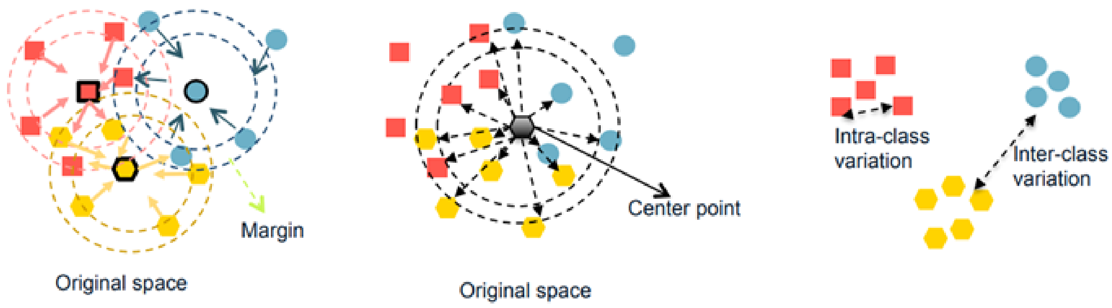


Figure 28 Illustration of intra class and inter class variation

In our cases, it fails to achieve the desire accuracies. From the plot we can see siamese perform bad. The reason for poor perform is that, this techniques is bad for inter-class variance and good for the intra class variance. Talking with a particulary cifar 10 dataset we have observed almost all algorithms perform poor and one of the reason is low intra-class variance. Our meta learning strategy can successfully discriminate between various classes if the inter-class variation is large. As a result, we were able to achieve great results on the MNIST and Fashion MNIST datasets.

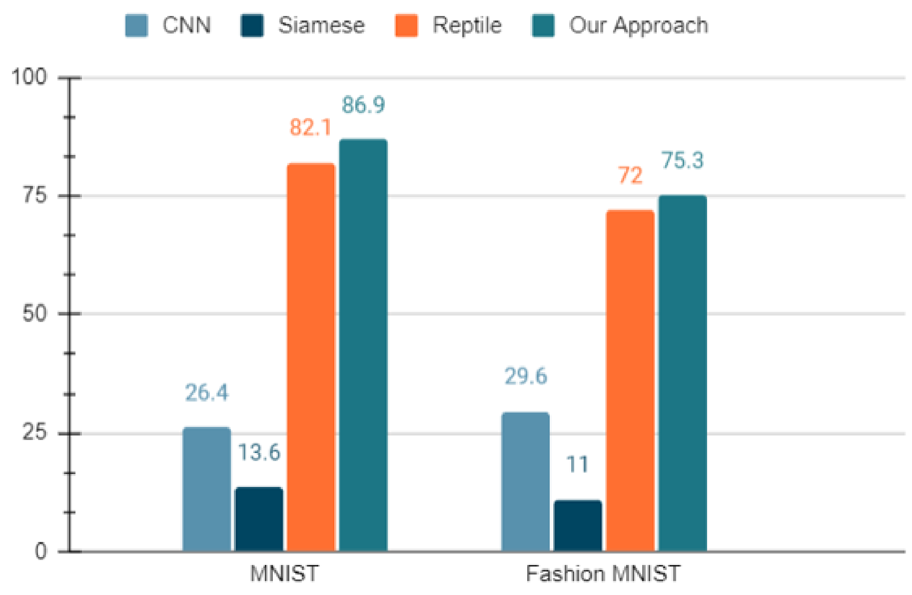


Figure 29 Grayscale dataset accuracy bar chart

In image recognition it is often assumed the method used to convert color images to grayscale has little impact on recognition performance. Fashion-MNIST and MNIST is a dataset comprising of 28×28 grayscale. This might be a reason for its good performance.

4.2 Resolution and quality of Data

A poor data quality is likely to reduce the performance of even an advanced, complex machine learner, while a good data quality can lead to high performance for a relatively simpler machine learner. Image quality is affected by different types of quality factors (such as resolution, noise, contrast, blur, compression). Resolution of the image affects the visual information of the image. Image with higher resolution contains more visual information details while an image with lower resolution contains fewer visual details. Convolutional neural network (CNN) based image classifiers always take input as an image, automatically learn image features, and classify into its output class. If input image resolution varies, then it hinders classification performance of CNN based image classifier. It is observed that this might be another factor Achieving less accuracies in Cifar-10 dataset because image resolutions in not up to mark.

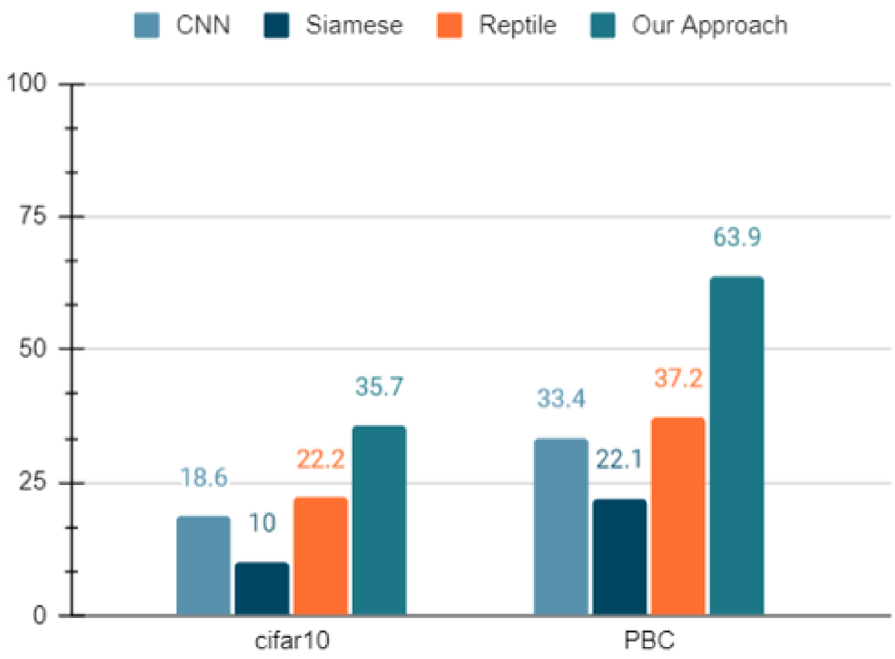


Figure 30 Colored dataset accuracy bar chart

CHAPTER 5

Conclusion

5.1 Conclusion

Few-Shot Learning (FSL) aims to bridge the gap between human learning and artificial intelligence. As artificial intelligence is capable of learning about new tasks comprising only a few examples with controlled data by combining preceding knowledge. We begin by formally defining FSL, following the relatedness and differences between conventional learning and meta learning approaches with FSL. We have developed a simple meta-learning algorithm named as stomatopods which is capable of learning on very few examples. In this research we provide a comparison of the accuracy achieved by our approach with various machine learning approaches such as conventional neural network, triplet network and reptile a meta learning model and proofs that our approach gives better results using a few examples. We then point out the core issue of convolution neural network and Siamese models learning with very few examples. We also give possible paths on problem settings, approaches, and application to explore in order to motivate future FSL research.

REFERENCES

[1] Z. Zhao and H. Liu, Public.asu.edu, 2007. [Online]. Available: http://www.public.asu.edu/~huanliu/papers/icml07.pdf.

[2] X. Li, J. Yan and J. Wu, "Anti-Noise Relation Network for Few-shot Learning - Enlighten: Publications", Eprints.gla.ac.uk, 2020. [Online]. Available: http://eprints.gla.ac.uk/225913/.

[3] Y. Wang, Q. Yao, J. Kwok and L. Ni, "Generalizing from a Few Examples: A Survey on Few-Shot Learning", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1904.05046.

[4] H. Ye, H. Hu, D. Zhan and F. Sha, "Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions", arXiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1812.03664. [Accessed: 7- Aug- 2021].

[5] C. Yang and S. Lim, "One-Shot Domain Adaptation For Face Generation", arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2003.12869. [Accessed: 16- Aug- 2021].

[6] J. Snell, K. Swersky and R. Zemel, "Prototypical Networks for Few-shot Learning", arXiv.org, 2017. [Online]. Available: https://arxiv.org/abs/1703.05175. [Accessed: 16- Aug- 2021].

[7] J. Schmidt, M. Marques, S. Botti and M. Marques, "Recent advances and applications of machine learning in solid-state materials science", 2019. [Online]. Available: https://www.nature.com/articles/s41524-019-0221-0. [Accessed: 16- Aug- 2021].

[8] M. North and C. Macal, "Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation", Oxford Scholarship Online, 2007.Available:

https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780195172119.001.0001/acprof-9780195172119

[9] T. Jerez and W. Kristjanpoller, "Effects of the validation set on stock returns forecasting", En.x-mol.com, 2020. [Online]. Available: https://en.x-mol.com/paper/article/1232148519321620480.

[10] Y. Wang, Q. Yao, J. Kwok and L. Ni, "Generalizing from a Few Examples: A Survey on Few-Shot Learning", arXiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1904.05046.

[11] G. Hinton, O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network", arXiv.org, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531. [Accessed: 16- Aug- 2021].

[12] O. Bohdal, Y. Yang and T. Hospedales, "Flexible Dataset Distillation: Learn Labels Instead of Images", arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2006.08572.

[13] Y. Wang, Q. Yao, J. Kwok and L. Ni, "Generalizing from a Few Examples: A Survey on Few-Shot Learning", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1904.05046.

[14] I. Sucholutsky and M. Schonlau, "'Less Than One'-Shot Learning: Learning N Classes From M<N Samples", arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2009.08449.

[15] Trung, L., 2019. https://euroasia-science.ru/pdf-arxiv/the-controllability-function-of-polynomial-for-descriptor-systems-23-31/. EurasianUnionScientists, 4(65).

[16] Y. Wang, Q. Yao, J. Kwok and L. Ni, "Generalizing from a Few Examples: A Survey on Few-Shot Learning", arXiv.org, 2019. [Online]. Available: https://arxiv.org/abs/1904.05046.

[17] Y. Roh, G. Heo, and S. E. Whang, “A survey on data collection for machine learning: A big data - AI integration perspective,” IEEE Trans. Knowl. Data Eng., vol. 33, no. 4, pp. 1328–1347, 2021.

[18] T. Wang, “DATASET DISTILLATION,” Arxiv.org. [Online]. Available: http://arxiv.org/abs/1811.10959v3.