Utilization of DCT Coefficients for the Classification of CIFAR-10 & CIFAR-100 Datasets

Author

LAIBA NAEEM CHAUDHARY

MS-17-CE-00000205029

Supervisor

Dr. Farhan Hussain

DEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

NOVEMBER, 2020

Utilization of DCT Coefficients for the Classification of CIFAR-10 & CIFAR-100 Datasets

Author

LAIBA NAEEM CHAUDHARY

MS-17-CE-00000205029

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

MS Computer Engineering

Thesis Supervisor:

Dr. Farhan Hussain

Thesis Supervisor's Signature: ____

DEPARTMENT OF COMPUTER AND SOFTWARE ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

NOVEMBER, 2020

Declaration

I certify that this research work titled "*Utilization of DCT Coefficients for the Classification of CIFAR-10 & CIFAR-100 Datasets"* 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.

Signature of Student
LAIBA NAEEM CHAUDHARY
MS-17-CE-00000205029

Language Correctness Certificate

This thesis is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the University for MS thesis work.

Signature of Student
LAIBA NAEEM CHAUDHARY
MS-17-CE-00000205029

Signature of Supervisor
Dr. Farhan Hussain

Copyright Statement

ACKNOWLEDGEMENTS

I am very thankful to my Creator **Allah Tala** for His ultimate guidance throughout this research work and for every new thought which He setup in my mind to improve it. Indeed, I could have done nothing without His priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was His will, so indeed none be worthy of praise but Him.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor **Dr. Farhan Hussain** for his help throughout my thesis and also for Advance Computer Architecture course which he has taught me. Despite the never ending commitments, he did never mind giving his maximum whenever I requested for his time and support.

I would also like to thank **Dr. Ali Hassan**, and **Dr. Arsalan Shaukat** for being on my thesis guidance and evaluation committee.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Dedicated to my exceptional parents, family and friends whose tremendous support and cooperation led me to this wonderful accomplishment

Abstract

In this work we propose efficient deep neural networks for classification that are well suited to the edge computing and cloud computing environment. These environments inherently have to deal with bandwidth limitations and bounded computational resources. Our proposed methods tend to reduce the bandwidth requirements and reduce the computational costs for running these deep learning algorithms. As an extensively used image compression algorithm, DCT (Discrete Cosine Transform) is used to reduce image information redundancy because a limited number of DCT coefficients can preserve the most significant image information. In this thesis, a novel frame work is presented by joining DCT coefficients and deep neural networks. We have targeted the deep neural network that can predict the most important DCT coefficients for an image and we have utilized those important DCT coefficients for classification purpose. As part of its training process, the proposed DCT model eliminates the input information which mostly represents the high frequencies. After achieving the important DCT coefficients from images we have applied the classification models on the compact representation of Grey scale and RGB datasets (MNIST, CIFAR-10 and CIFAR-100). We have used two approaches for classification first is classification by important DCT coefficients and second is classification by low resolution images. We have used VGG-16 and purposed CNN architecture for classification purpose. Additionally we have also proposed the prediction model for predicting the important DCT coefficients by using Multi-Layered Perceptron (MLP) model. The experiments has shown the promising results and we have found out that we can achieve almost the same classification accuracy with compact representation of Grey Scale and RGB datasets as we were achieving with raw pixels.

Key Words: *Discrete Cosine Transform, Image Compression, DCT, CNN, Deep Learning, VGG-16, MLP, Zigzag Scanning, JPEG Compression, Classification.*

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

CHAPTER 1: INTRODUCTION

In this chapter, the detailed introduction about the research work is presented and it is divided in to following sub sections. **Section 1.1** presents the problem statement and gives a brief introduction of our work. **Section 1.2** discusses various approaches and techniques that are relevant to our solution. These include different methods for efficient image and video compression. Research contribution and thesis organization is given in **Section 1.3** and **Section 1.4** respectively.

1.1 Overview

Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data. These deep learning architectures have now a day become the state of art algorithms for classification, prediction, function approximation, segmentation, localization, regression, and detection problems. Although these algorithms provide excellent results on image and video data they need to process enormous amount of information for any of the above mentioned tasks. Correspondingly processing huge amount of data means more computationally expensive networks.

Another challenge arises due to the increased progress in Internet of Things, the number of smart devices that are connected to internet is growing, which results in large scale data and causes issues in bandwidth load. Today's diverse needs of data processing is no longer sufficiently handled by conventional cloud computing, therefore edge computing technologies have taken this responsibility[1] [2] [3]. However in an edge computing environment data needs to be transferred from the source node to the computing (Edge) node. These bandwidths requirements can become very challenging in case of image and video data[4]. A highly researched area nowadays is the reduction in cost of bandwidth during data transmission in edge computing.

In this work we aim to target the classification of standard image datasets by deep learning models but with much less input information. Less number of input information corresponds to less number of input neurons and less number of network connections. This generally reduces the overall computational cost of the network. Also if we are targeting this image classification problem in an edge computing environment then by using lesser input information for classification we can very well address the bandwidth issues in such an environment. A small amount of image information needs to be transmitted to the edge device in such a setting thus making them able to efficiently work in low bandwidth framework. Also a network with low computational cost (due to lesser input) would be well suited to an edge device. In our work we tend to explore the use of discrete cosine transforms [5] for compact representation of images. The next section defines a few the image compression algorithms in general with particular emphasis on discrete cosine transforms.

1.2 Background

These days many other methods are being addressed for reducing the data size for example image compression is also used for the same purpose it is used to provide the compact representation of images. The compression of images or videos can be achieved by selectively ignoring the unimportant part of the information [6]. There are various algorithms which are used for image compression, Most of the conventional lossless image compressions are based on prediction compression which predicts the value of current symbol by using several symbols which have already been encoded [7]. For example, image compression by prediction method [8]in which hierarchical prediction techniques are used to for horizontal, vertical and diagonal predictors to predict the pixels. It removes the prediction error rate near edges and preserves the sharpness of the images. Wavelet transform based methods are also used for images and videos compression. Wavelet transform decomposes an image or signal into a set of different basis functions and these basis functions are called wavelets. It presents the mathematical way of encoding process in such a way that it is layered according to level of details [9]. JPEG is a widely used standard image compression technique based on DCT. JPEG is done in two ways, first is lossy image compression and second is lossless image compression. In the lossless image compression the reconstructed image matches exactly with the original image. This type of the compression is usually takes place in case of medical images and legal record where we cannot afford of any loss of data. While lossy image compression techniques are very useful in video conferencing, TV broadcast and facsimile transmission. In such scenarios, some amount of error is acceptable. Such type of compression algorithms are usually more complex and needs more computations than lossless image compression [10]. In literature, we found that lossy compression techniques are most preferable as they much capable of achieving better compression [11] [12].

Nowadays, DCT is extensively used in image and video compression [13]. The Discrete Cosine Transform is a process for transforming a signal or image in to its frequency components [14], it converts the image in to sum of sinusoids having different magnitudes and frequencies. The definition of the two-dimensional DCT for an input image P and output image Q is.

$$Q_{xy} = \alpha_x \alpha_y \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} P_{mn} \, cos \frac{\pi(2m+1)x}{2M} \, cos \frac{\pi(2n+1)y}{2N}, 0 \le x \le M-1, 0 \le y \le N-1 \tag{1.1}$$

Here,

$$\alpha_x = \begin{cases} \dfrac{1}{\sqrt{M}}, & x = 0 \\ \sqrt{\dfrac{2}{M}}, & 1 \le x \ \le M-1 \end{cases}$$

And

$$\alpha_y = \begin{cases} \dfrac{1}{\sqrt{N}}, & y = 0 \\ \sqrt{\dfrac{2}{N}}, & 1 \le y \le N-1 \end{cases}$$

M and N are the row and column size of P, respectively.

Similarly, the two dimensional Inverse Discrete Cosine Transform is defined as:

$$P_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \alpha_x \ \ \alpha_y \ Q_{xy} cos \frac{\pi(2m+1)x}{2M} \, cos \frac{\pi(2n+1)y}{2N}, \ 0 \le m \le M-1 \, 0 \le n \le N-1 \tag{1.2}$$

Here

$$\alpha_x = \begin{cases} \dfrac{1}{\sqrt{M}}, & x = 0 \\ \sqrt{\dfrac{2}{N}}, & 1 \le x \le M-1 \end{cases}$$

And

$$\alpha_y = \begin{cases} \frac{1}{\sqrt{N}}, & y = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq y \leq N-1 \end{cases}$$

Here M and N are the row and column size of the input and output images respectively.

For more understating of this topic conversion of an image from spatial domain to frequency domain is shown in **Figure 1.1**. For the image of size AXB, f (i, j) represents the intensity of pixel at x (i, j), while DCT coefficient for the pixel x (i, j) is represented by F (u, v). The output matrix of DCT coefficients contains integers, which lies from -1024 to 1023. As most of the matrix energy lies in low frequencies which appear in the left corner of DCT matrix, so we can discard the high frequencies which are small enough to be neglected.

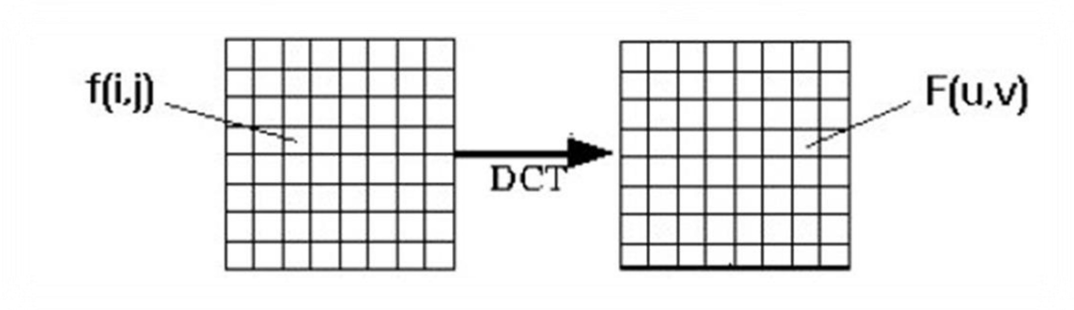


Figure 1.1: Image Transformation from Spatial Domain to Frequency Domain

One of the most important features of FDCT is that it focuses heavily on signal strength on a few converted DCT coefficients at low spatial frequencies. In other words, the number of DCT coefficients of very large size is very low and small coefficients that are far are large in value. More often than not, much of the detail in the image is reflected in low frequencies. High frequencies often involve sharp changes that add extremely fine detail to the image.

It can be seen in the **Figure 1.2(a)** that basis functions are exhibiting a continuous increase in both the horizontal and vertical frequency. It can be notice in **Figure 1.2(c)** that the DCT coefficients having the low frequency components representing the high energy are mostly located in the upper left corner of the DCT matrix. As described in **Figure. 1.2(d)** zigzag scan

can be used to group low frequency components in top of vector. The zig zag scan is used to map 32x32 in to 1x1024 vector and to gather low frequency coefficients present at the top of the vector. We have performed the series of experiments for the selected DCT coefficients and found the classification accuracy for each DCT selection we made on 32x32 and 8x8 block of data.

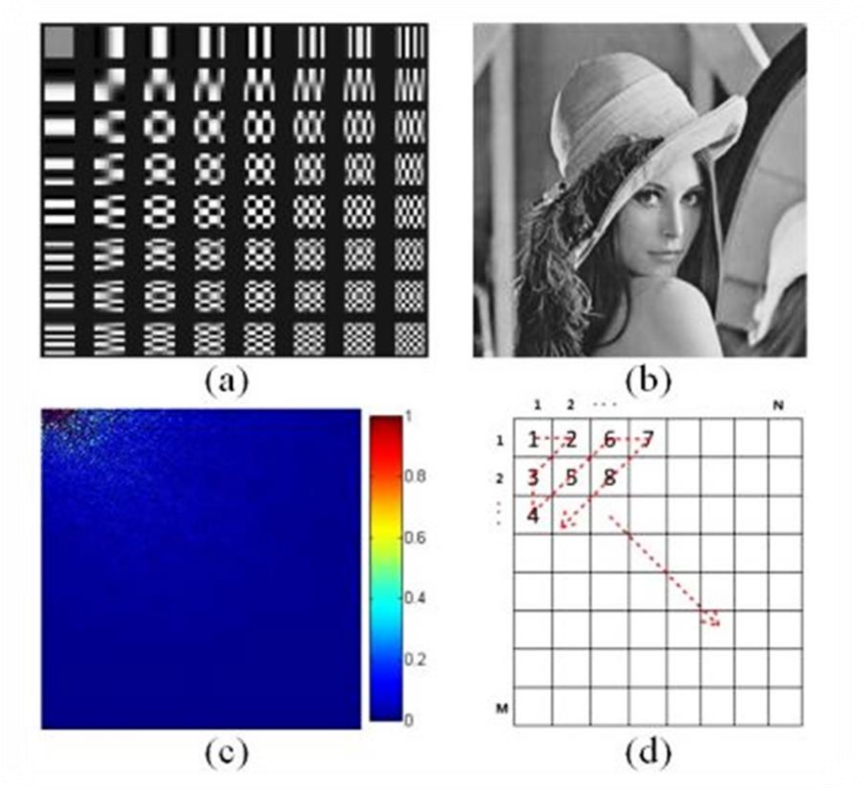
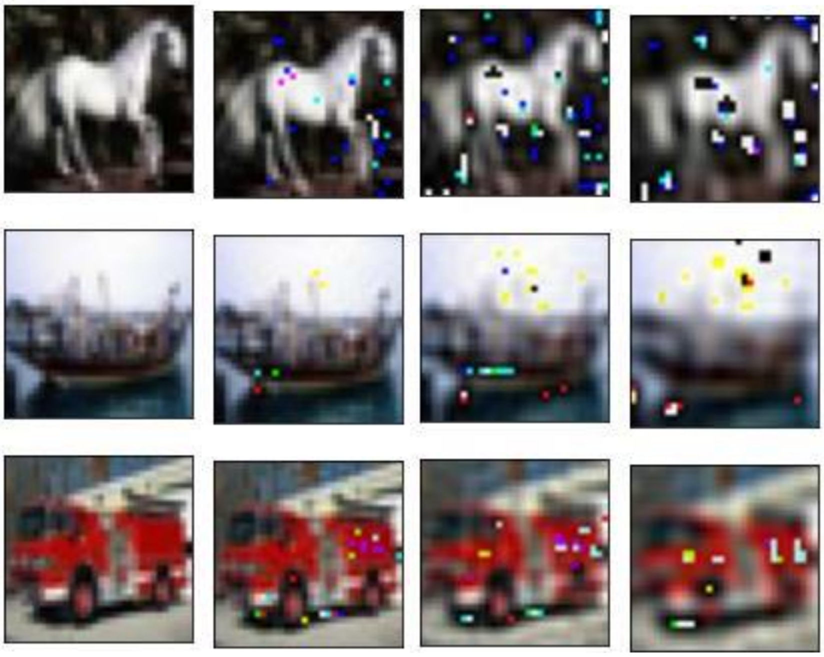


Figure 1.2: DCT Basis Functions of Image Lena and Zigzag scanning

Fig 1.2 **(a)** shows the 2-D DCT basis functions for 8x8 matrix. **(b)** Represents the original image. **(c)** DCT coefficients matrix of (b) **(d)** zigzag pattern used to select important DCT co-efficient

Figure 1.3 depicts the original images and their compressed versions. It can be seen that as we decrease the number of important DCT coefficients we begin lose the image quality as well as resolution of the images is reduced.



(a) (b) (c) (d)

Figure 1.3: (a) Original Images **(b)** Reconstructed Images by 60% DCT Coefficients **(c)** Image Reconstructed Images by 25% DCT Coefficients **(d)** Reconstructed Images by 15% DCT Coefficients

1.3 Research Contributions

The major contributions from this research are as follows:

- We designed a neural network that was able to predict the most important DCT Coefficients of an image.
- We achieved classification of standard image datasets with several deep learning architectures, requiring much less information to perform this classification task. The amount of information was reduced by
    I. By using low resolution images for classification purpose
    II. By using only the important DCT coefficients for classification.

All the above mentioned techniques and algorithms were evaluated on standard image data sets starting from the MNIST and extending our work to more complex datasets like CIFAR 10 and CIFAR 100.

1.5 Thesis Organization

This document consists of 5 chapters. **Chapter 1** explains the detailed introduction of research problem and it consists of overview, problem statement, research flow, research contributions, and thesis organization. **Chapter 2** gives the comprehensive literature review focusing the existing work done in the field of DCT domain by various researchers and scholars. **Chapter 3** provides the overview of the proposed methodology, and classifiers used in this study. Flow diagrams represent the flow of proposed methodology. **Chapter 4** provides the details about experimentations, results and details about datasets. **Chapter 5** provides the discussion of the whole thesis with the limitations and finally **Chapter 6** gives the conclusion of the research work and suggests the future work.

Chapter 2

Literature Review

CHAPTER 2: LITERATURE REVIEW

In this chapter we present a brief overview of techniques and methods that are related to reducing the computations performed by a deep neural network. We also discuss deep learning algorithms that employ the use of Discrete Cosine Transforms for classification purpose. After the brief literature review of the studies from known databases we have mentioned the research gaps present in earlier studies.

2.1 Related Work

In literature standard techniques have been purposed for reducing the complexity of neural networks such as Factorization and decomposition of the convolution kernel. [15] Proposed to factorize the convolutional neural networks to reduce its computations and introduced the topological subdivisions in order to lessen the connections between input and output channels. [16] Proposed the model compression for deep CNNs by conducting the filter selection and filter learning at the same time. A factorized convolution filter which contains the standard real valued convolution filer and a scalar has been proposed. [17] Offers the scheme which is based on the tensor decomposition in order to accelerate the convolution neural networks. The given method removes the linear redundancy in convolution kernels and also highly speeds up the CNNs and maintaining the high classification accuracy at the same time. The results show that the proposed scheme is able to achieve the speed of whole model by x4 with 1.9% increase in top-5 error in AlexNet.

Recently, the use of the separable convolution kernels in deep neural network architectures has been discussed. Different researchers [18] have used this techniques in their deep architectures and have achieved state of the art or near to state of the art performance. [19] Focuses on the real time implementation of a separable convolution kernel which is based on distributed arithmetic having support for handling high resolution images by using large kernel size. This proposed architecture processes the pixels directly when available from external memory which results in achieving the low processing time as compared to existing works.

Quantization is also used for reducing the complexity of deep neural networks. Such as [20] uses RNN (recurrent neural network) decoder with code book using weight quantization. Test results shows that memory overhead can be reduced by 98% and it also reduce the computational complexity with a minor performance loss. [15] Introduces the novel technique in order to train the low bit networks using weights and activation quantized different bits. It also addresses the two issues, first is approximate activations from low bit discretization to reduce network overall computational cost and dot product memory and the second is to mention the weight quantization and provide mechanism for discrete weights to avoid gradient mismatch. [16] Purposes the new method for simplification of a trained deep neural network by finding an optimal, unique and optimal precision for each network parameter in which increase in loss in minimized. The purposed method was experimented on CIFAR, MNIST and SCHN datasets and results shows that near the state of the art reduction in model size is achieved.

Pruning methods have also been used to cater the network complexity problems. In [21] author uses the sparse decomposition to reduce the redundancy of parameters in deep neural networks. Using the purposed procedure, the author was able to zeros out above 90% of the parameters having only a drop of less than 1% accuracy on ILSVRC2012 dataset. As structural pruning of a neural network lessons the energy, memory transfer and computation costs during inference. Therefore [22] proposed the method that calculates the involvement of neuron to a final loss and then removes those neurons with littler scores.

Deep learning is mainly used for images and videos processing tasks. Nowadays, as digital imaging is gradually developing therefore, large stream of images are expected to be produced. The image size can be a problem keeping in mind the two specific scenarios, one is required bandwidth capability and second is storage space needed by those large streams of visual data. The transmission bandwidth and the storage requirement needed by the uncompressed data is significant [23]. In order to cope up with bandwidth and storage requirements various work in image compression techniques have been done and important achievements have been gained in the area of reducing storage requirements and bandwidth capacity[24] [25] [26]. DCT can e used for image and video compression techniques. In our thesis, we have focused on Discrete Transform which is a core component in the image compression. DCT has a close connection with Discrete Fourier Transform, it is basically the extended version of DFT [27]. As Discrete Fourier transform signals are periodic and therefore, most of the times discontinuities are found at the boundaries. Any arbitrary value taken from segment of the signal does not contain the same value at its left and right both boundaries. While Discrete Cosine Transform signals can have more coefficients and thus keep the required shape of the truncation [28] and it shows that DCT is most preferable on DFT in case of image processing tasks. DCT are mainly used in Joint Photographic Experts methods, also known as JPEG compression techniques. JPEG uses both lossy and lossless image compression techniques on images and video data. A lot of work has been done in compression and classification from DCT coefficients. Also many research papers have discussed the problem of image recognition and classification by the use of Discrete Cosine Transform coefficients by applying different techniques. **Table 2.1**, shows the most relevant present in the literature.

Table 2.1: Summary of carefully chosen studies according to scientific databases and publication type.

Literature	Publisher	Year	Type	Description
[29]	**IEEE**	2020	Journal	DCT based CNN is used for the detection of Median Filtering Forensics
[30]	**IEEE**	2019	Conference	Compressed DCT coefficients from JPEG compression are used for classification
[31]	**IEEE**	2019	Conference	Proposed CNN-C and CNNRC3 for classification from DCT coefficients
[32]	**IEEE**	2019	Conference	DCT-Net is proposed for Classification of noisy/blur Images
[33]	**Elsevier**	2020	Journal	DCT based CNN has been proposed to reduce JPEG Compression Artefacts
[34]	**IEEE**	2014	Conference	Auto Encoder using DCT coefficients is proposed for classification tasks
[35]	**ACM**	2018	Conference	Using DNN a JPEG compression framework has

				been presented for ImageNet dataset
[36]	**ACM**	2018	Conference	Resnet-50 has been used DCT coefficients available in the middle of JPEG Codec
[37]	**IMVIP**	2017	Conference	Classification using Proposed CNN has been presented using DCT coefficients
[38]	**Springer link**	2019	Journal	DCT based colour image compression algorithm has been presented using Adaptive block Scanning
[39]	**IEEE**	2019	Conference	A CNN model has been proposed from classification for detection of compressed JPEG images
[40]	**IEEE**	2016	Conference	DCT operation is performed on the feature maps generated by the layers of CNN
[41]	**IEEE**	1997	Conference	ANN has been used for image compression and DCT computation
[42]	**IEEE**	2018	Conference	CNN has been used for classification of embedded systems
[43]	**NIPS**	2012	Conference	1.2 million high resolution images of ImageNet has been classified using Deep CNN
[44]	**IEEE**	IEEE	Conference	Cifar-10 Classification has been proposed using Deep CNN

Using Neural Networks, a lot of searches have been done in image processing domain [45]. The very first approaches for using Neural Networks in image compression tasks are mentioned in [46] [47] [48] [49]. Nowadays, Convolution Neural Networks are also becoming very popular in image classification [50], computer vision, image segmentation [51] and recognition [52] and understanding [53] [54] [55]. Normally, image classification is performed either supervised or unsupervised algorithms. A supervised image classification method on underwater fish detection is presented in [56] and a supervised classification algorithm is presented in [57]. In literature, DCT has been used in various researches for classification, detection and for identification purposed. In the above recent studies in DCT based classification have been presented. DCT transforms the each input data in to its frequency components [58] [59].

Learning from frequency data is always been considered in past [60] presents that only 35% of DCT coefficients are enough for recognition of the image as face. [61] [62] shows the DCT on EEG and it focuses on using low frequency components and to avoid the high ones as most of the signal information is present in the low frequency data. As ImageNet challenge [63] has encouraged the new number of image classification models which starts from AlexNet [43], ResNet [64], ZFNet [65], VGG-NET [66], GoogleNET [67]. VGG-16 that uses Convolution having small sized filters have been used in literature and we have also used the VGG-16 [68] architecture in our proposed approach and found the results which gives better results as compared to previous approaches mentioned in above table. After a brief literature review and analysis we have targeted the classification accuracy of [23] and our results prove that we have gained the more accuracy as mentioned in this paper for CIFAR-10 and CIFAR-100 datasets.

Chapter 3

Proposed Methodology

CHAPTER 3: PROPOSED METHODOLOGY

The problem statement of our thesis revolves around the classification of standard image datasets by deep learning architectures while working in an edge computing environment. The major challenges in such an environment are the limitations on the

1) Bandwidth requirements (for transferring data to the edge device)

2) Computational capabilities of the edge device

In case of videos and images the above challenges can become more intensified. We propose two methods as a solution to the above challenges while targeting the image classification problem.

Both of these methods broadly consist of transforming the images to frequency domain by using the Discrete Cosine Transform. The cost of bandwidth requirements is mitigated by transmitting only a subset of these DCT coefficients. These methods differ in how the classification is carried out at the edge device i.e.

1) Classification is carried out by employing only the significant low energy coefficients

2) Classification is carried out by reconstructing low resolution images from these reduced coefficients.

In the first section of this chapter we present the conventional way of transforming an image in to frequency domain by employing Discrete Cosine Transforms and how we can extract the most significant values of the DCT for an image. In the 2nd and 3rd sections we discuss both of our proposed solutions along with the respective deep learning architectures in detail.

3.1 Extraction of DCT from an Image

Discrete cosine transform possess the property that, for a particular image, most of the high energy components, containing the most visually significant information of the image concentrated in just a few DCT coefficients.

In this research zigzag scan has been used on each 8x8 and 32x32 matrix starting from upper left corner and then convert it in to a one dimensional DCT coefficient vector of 1x64 and 1x1024 respectively. **Figure 3.1** shows the zigzag scan of an 8x8 matrix. As DCT converts the input image in to linear combination of varying frequency components and mostly components are not large in magnitude. Therefore, visually significant details about the image are present in just a few coefficients. Hence, we can remove the high frequency components so that amount of data needed to describe the image can be reduced as it does not disturb the too much image quality [25].

.

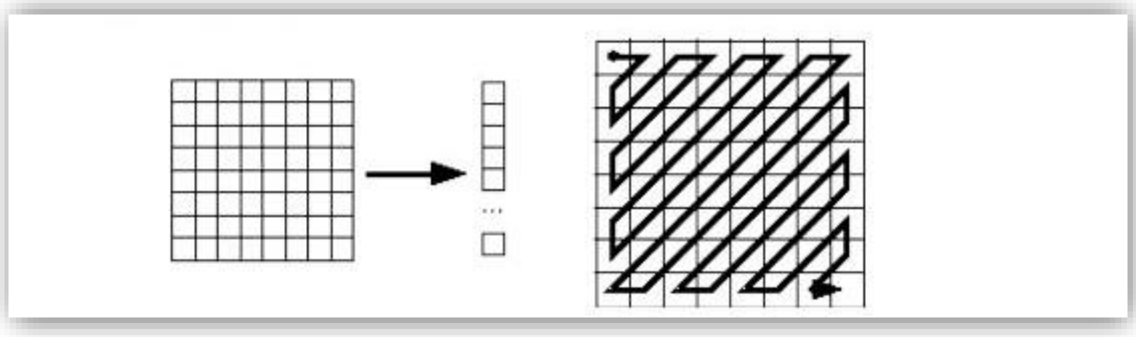


Figure 3.1: Zigzag Scanning

The selection of input size being sent to the DCT network has been done by two approaches. In our first approach we have taken the input size equal to image size of the dataset and in the second approach we have divided the image in equal 8x8 blocks. Discrete cosine transform is then performed on both the inputs. Once we get the DCT transform of an image we can address the challenge of limited bandwidth by transmitting only a subset of these values.

3.2 Classification from the Bare DCT coefficients

In this section we present our first approach of classifying the standard image datasets. In this approach we only utilized the bare DCT coefficients for classification purpose. We broadly designed networks with two different approaches. In the first approach we designed the network such that it classifies the images based on the DCT of the whole image. In the second approach the network is designed such that it classifies the images based on the DCT of 8*8 blocks. These networks are discussed in section 3.2.1 and 3.2.2 respectively.

3.2.1 Classification using Whole Image as Input to DCT

Same above technique was applied by taking full image as input to DCT instead of dividing it in to blocks and after applying the Zigzag scan to them and selecting high energy spatial frequencies, zero padding in place of discarded coefficients was done. Thus, in this method matrix equal to image length is regenerated by placing blocks in to its location that is the input to CNN and VGG-16. **Figure 3.2** represents this approach in the form of flow diagram.

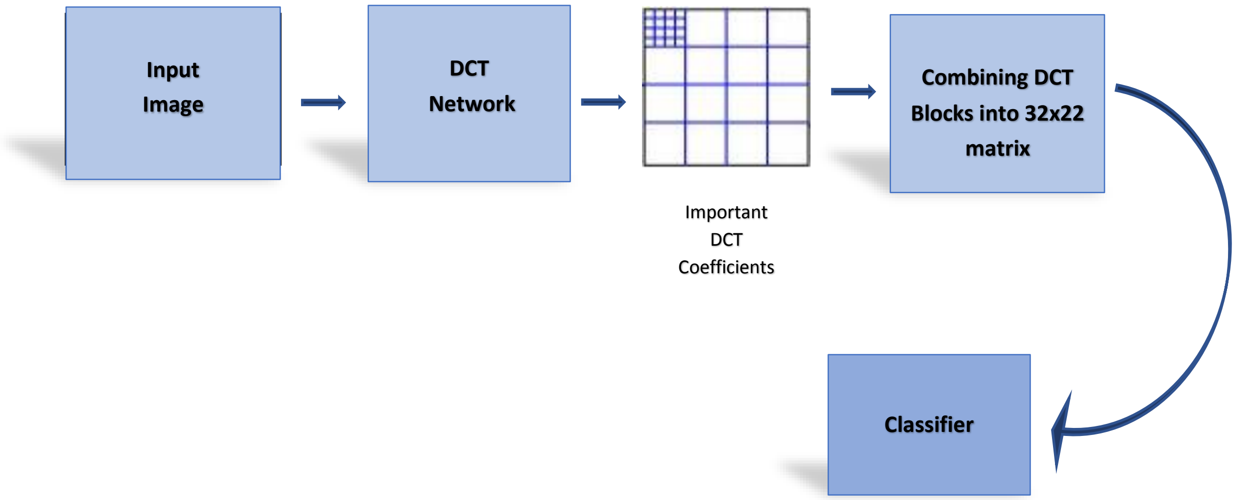

Input Image
DCT Network
Important DCT Coefficients
Combining DCT Blocks into 32x22 matrix
Classifier

Figure 3.2: Classification from DCT coefficients by selecting important DCT coefficients from Whole Image

3.2.2 Classification using 8x8 Block as Input to DCT

In the second approach, instead of taking the inverse discrete cosine, input to the CNN and VGG-16 is the important DCT coefficients containing the high energy information of the image. As it can be seen in **Figure 3.3**, each image is first divided in to 8x8 non overlapping blocks, these blocks are input to DCT. After applying the Zigzag scan to these blocks and selecting high energy spatial frequencies, zero padding in place of discarded coefficients was done. Thus, in this method matrix equal to image length is regenerated by placing blocks in to its location.

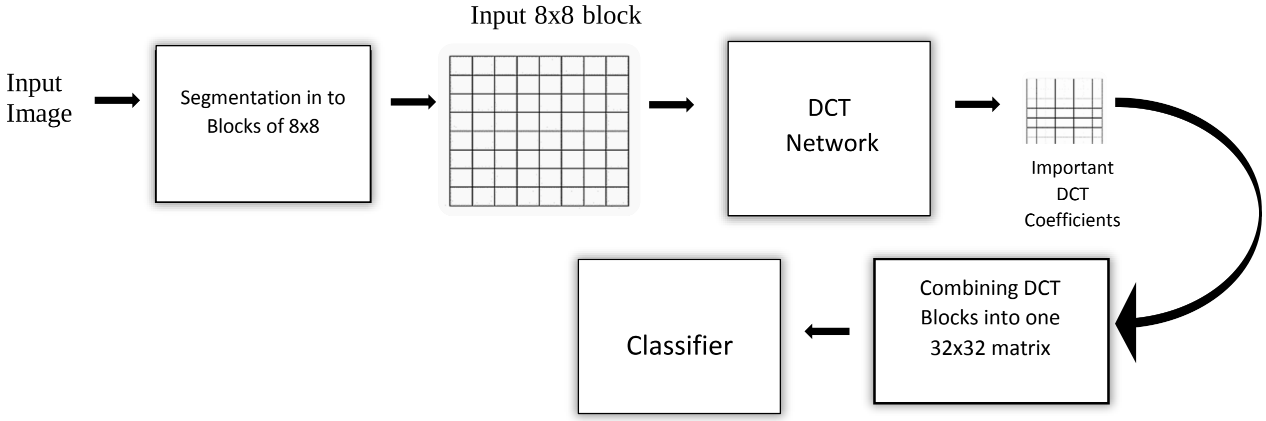

Input 8x8 block
Input Image
Segmentation in to Blocks of 8x8
DCT Network
Important DCT Coefficients
Combining DCT Blocks into one 32x32 matrix
Classifier

Figure 3.3: Classification from DCT coefficients by selecting important DCT coefficients from 8x8 block

These approach not only saves the bandwidth requirement but also reduces the complexity of the network because we are only utilizing a few number of DCT coefficients (as the input to the neural network is vastly decreased so does the complexity of the neural net also decrease).

3.3 Classification from the Low Resolution Images

In this section we present our second approach of classifying the standard image datasets. In this approach we reconstruct the low resolution images from the subset of transmitted DCT coefficients and then utilize these low resolution images for classification purpose. Once again keeping the uniformity with section 3.2 we broadly designed networks with two different approaches. In the first approach we reconstructed low resolution images from the subset of DCT coefficients which corresponded to the DCT taken of the whole image. In the second approach the low resolution images were constructed from the subset of DCT coefficients which corresponded to DCT taken when the original image was divided in to 8*8 blocks. These approaches are discussed in section 3.3.1 and 3.3.2 respectively.

3.3.1 Classification using Whole Image as Input to DCT

Similarly, instead of dividing the image in to smaller blocks the input to DCT is one block with the dimension similar to the height and width of the original input image. DCT takes the input and outputs the corresponding basis functions or DCT coefficients. Thus, the DCT coefficients obtained can be regarded as relative amount of 2D spatial frequencies present in the original input image. Zigzag scan is applied on the resultant DCT coefficients. As the most important features such as texture, complexity and uniformity are concentrated in a few transformed basis function having the lower spatial frequencies, hence we have investigated the classification accuracy by gradually increasing those lower spatial frequencies components and discarding the high frequency components. Zero padding is done in place of discarded high frequency components and then IDCT of these low spatial frequencies was taken which results in low resolution images. Experimental results shows that we can achieve an excellent amount of accuracy with a small subset of DCT coefficients. Flow diagram of purposed approach is shown in **Figure 3.4**.

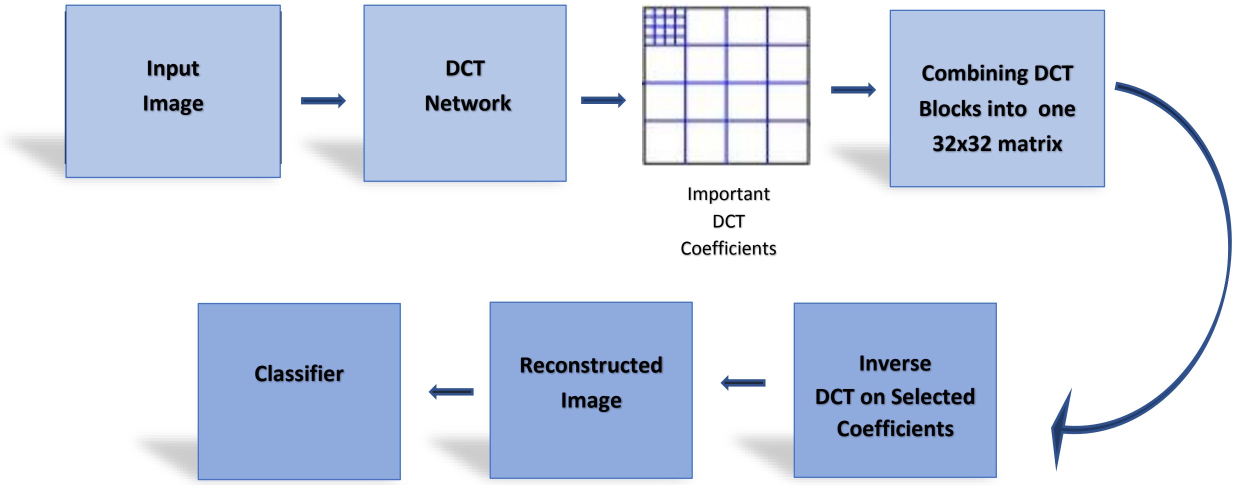

Input Image
DCT Network
Important DCT Coefficients
Combining DCT Blocks into one 32x32 matrix
Inverse DCT on Selected Coefficients
Reconstructed Image
Classifier

Figure 3.4: Classification from low resolution images by selecting important DCT coefficients from Whole Image

3.3.2 Classification using 8x8 Block as Input to DCT

In this approach each 8x8 block from image is transformed in to a discrete coefficient matrix of size 8x8. Zigzag scan is applied on the resultant DCT coefficients. As discussed above, it is observed that most important DCT coefficients containing the most of energy are in upper left corner of the discrete cosine transform matrix so we have investigated this approach by continuously increasing the DCT coefficients in each upper left corner and discarding the high frequency components in order to lessen the amount of data needed to describe the image. Zero padding is done in place of discarded high frequency components. Thus it again regenerates the 8x8 block. Inverse zigzag is performed on this block. Resultant block and flow diagram have been described in this **Figure 3.5.** Inverse discrete cosine is performed on the each resultant blocks and matrix equal to full image is regenerated that results in to low resolution image, which is the input to CNN and VGG-16.

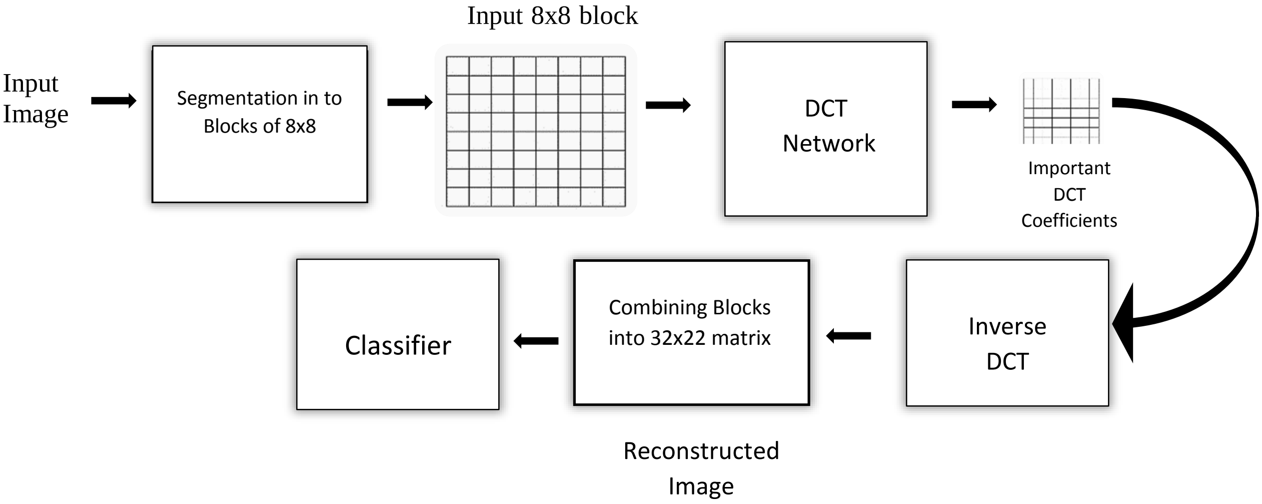

Input 8x8 block
Input Image
Segmentation in to Blocks of 8x8
DCT Network
Important DCT Coefficients
Inverse DCT
Combining Blocks into 32x22 matrix
Classifier
Reconstructed Image

Figure 3.5: Classification from low resolution images by selecting important DCT coefficients from 8x8 block

At the end of this section you should amplify that this approach only saves the bandwidth requirement it doesn't greatly reduce the complexity of the network because we are not decreasing the number of neurons in the input layer.

3.4 Neural Network for the Prediction of Most Important DCT Coefficients

Another contribution in this thesis is that we have proposed a neural network based on Multi-Layer Perceptron, this network can predict the important DCT coefficients for each 8x8 block. The input to this network is 8x8 block of image data and it outputs the important DCT coefficients from each block. This network is based on MLP which is class of feed forward artificial neural network. MLP models are designed to achieve the important 50, 40 and 30

DCT coefficients from each 64 coefficients in 8x8 block of data. Each MLP model consists of six hidden layers. The output layer contains the varying number of neurons which represents the DCT coefficients. The proposed prediction model is shown in **Figure 3.6**.

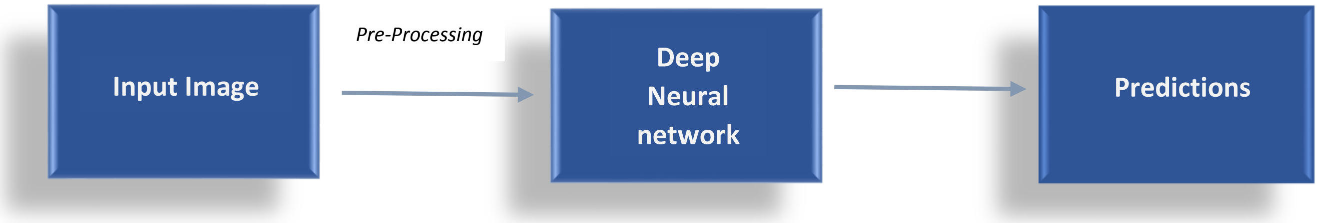


Figure 3.6: Proposed DCT Prediction Model using MLP

3.4.1 Proposed MLP Architecture

Cifar-10 and Cifar-100 datasets are used for training of DCT prediction model based on MLP. Each image in both datasets is divided in to blocks of 8x8 and then neural network was trained on these blocks. As in supervised learning, the desired output of the neural network is always known so important DCT coefficients from each block are used as labels. As it can be seen in **Figure 3.7**, input layer of DNN is fed with 8x8 blocks of original image and the output layer predicts the values of important DCT coefficients from each block. Four different estimates of important DCT coefficients i.e.

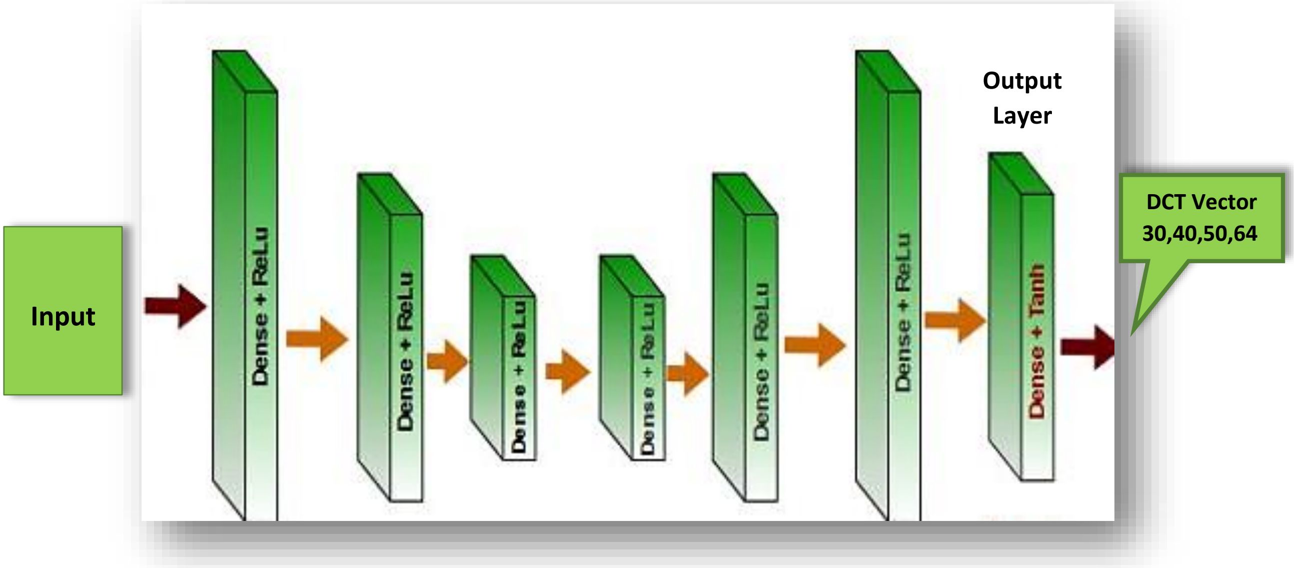


Figure 3.7: Proposed DCT Coefficients Prediction Model

30, 40, 50 and 64 are used for as label for given training set of images and each prediction neural network model was trained on 500 epochs

3.4.1.1 Activation Functions

In order to get understanding of activation functions, we first have to understand the functionality of an artificial neuron. Neurons estimate the weighted sum of their inputs and add bias to it and then take the decision of whether they should be fired or not. There are plenty of activation functions depends upon each case. In out proposed MLP model, we have used two different activation functions i.e. ReLU and tanh.

- **ReLU (Rectified Linear Units)**

ReLU [69] works by making sure that output does not become the negative value. Therefore, when x is greater than zero then the output stays x and if the goes below zero the output stays zero. In our network, ReLU is used at input layers in order to take a decision whether to fire neurons to hidden layers or not.

$$(\boldsymbol{x}) = \mathbf{max}(\mathbf{0}, \boldsymbol{x}) \tag{4.1}$$

ReLU is widely used for one of its main reasons that it does not activate the whole set of neurons at once. For example, when it gets the negative input it converts in to zero and neuron will not be activated. Which means, at a given time, only those neurons will be activated which fulfil the above criteria and in this way ReLU makes the Artificial Neural network sparse and hence helps in increasing its efficiency.

- **Tanh Function**

Tanh [70] activation function is used to limit the output in a range of (-1,1). In our proposed neural network we have used the Tanh in the output layer to predict the values of DCT coefficients.

$$\boldsymbol{f(z) = \tanh(z)} \tag{4.2}$$

Normally, Tanh is used when we have a regression problem just like in our case where we want to predict the normalized values of DCT coefficients. As the DCT coefficients values range from positive to negative therefore, we have used Tanh in the output layer.

3.4.1.2 Optimization

In our work, we have used adaptive moment estimation technique for weight optimization. Details are given below.

- **Adam**

Adam (Adaptive Moment Estimation) [71] is known for computing the adaptive learning rates for all the parameters in a neural network. Apart from saving the exponentially decreasing averages of earlier squared gradients, like Adadelta or RMSprop, it is also somewhat similar to momentum. If momentum is termed as a ball moving down a slope then Adam can be thought of heavy ball with large friction and hence giving us flat minima.

3.6.1.3 Loss Function

Loss functions in machine learning, are objective function that needs to be minimized. A loss function finds how close the predicted output and actual output is and how good the results of prediction models are. The purpose of our purposed neural network was to produce the DCT coefficients with as little error as possible.

- **Mean Absolute Error**

Mean Absolute Error [72] is a loss function which is used in regression models. It is the addition of absolute difference between the desired and actual variables. It calculates the mean magnitude of errors in the given set of predictions without considering their directions. We have used Mean Absolute Error instead of mean squared error as MAE is more vigorous to outliers and one high values cannot disturb the output.

$$\boldsymbol{MAE = \frac{\sum_{i=1}^{n}|y_i - y_i^p|}{n}} \qquad (4.3)$$

Chapter 4

Implementation and Results

CHAPTER 4: EXPERIMENTATION AND RESULTS

In this section, a brief overview of experimentation details and results of this thesis are discussed. This section is further divided in to sub parts. Section 4.1 provides the explanation of experimental setup, design assessment and give the brief information about datasets used in this research. Section 4.2 describes the configurations of the deep learning architectures employed in detail Section 4.3 and Section 4.4 presents the result of our proposed techniques while Section 4.5 gives the result of our DCT prediction network.

4.1 Experimental Evaluation

The proposed approaches are executed on GPU with Tensorflow deep learning framework [73]. The neural network design and training are implemented on Linux Based 64-bit Intel 4-core Xeon E5520 processors, 8 physical cores, 24GB DDR3 RAM. The programming language used is Python(3.5.2) having Keras framework [74] version 2.1.0, pandas, numpy, scipy.io, os, scipy.fftpack libraries in python are used for implementation. Cifar10 and Cifar-100 are downloaded from Keras Library. The total training for CNN model is about 4 hours and for VGG-16 model it almost takes 5 hours.

4.1.1 Design Assessment

In this thesis, we have purposed the two less complex, more robust approaches for classification of CIFAR-10 and CIFAR-100 datasets by utilizing the important DCT coefficients. For initial understanding and we have first implemented this approach on MNIST dataset which contains 60,000 training and 10,000 testing grey scale images containing hand written digits from 0 to 9. For classification we have used CNN and VGG-16 and for DCT Prediction Model we have used Multi-Layer perceptron neural network. Most important DCT coefficients are used as reduced representation of the each image in both datasets and hence we have achieved the same classification accuracy from this reduced representation same as we were getting from raw pixels.

4.1.2 Datasets

For initial understanding we have performed the purposed approaches on Modified National Institute of Standards and Technology Database i.e.; MNIST [75]. Image dataset consists of 70,000 grey scale images of handwritten digits between 0 and 9 has been used. The dimension of the each image was 28 x 28 and pixel value of the images falls in the range between 0 and 255. After achieving successful experimental results on MNIST we extended out approaches to CIFAR-10 and CIFAR-100 datasets[76]. CIFAR-10 contains 60,000 RGB images of size 32x32x3 of 10 classes having 6,000 images in each class. CIFAR-100 has 100 classes which contains 600 images in each class, containing 500 RGB training images and 100 RGB testing images in each class. Dimensions of CIFAR-100 dataset is same as CIFAR-10. Overall split in both datasets is 50,000 training and 10,000 test images. In order to validate the results each dataset was divided in to two parts i.e.; training and testing. In MNIST dataset 60,000 images were used as training purpose and 10,000 were used as testing purpose. While in case of CIFAR-10 and CIFAR-100 datasets 50,000 images were used for training and 10,000 images were used for testing purpose.

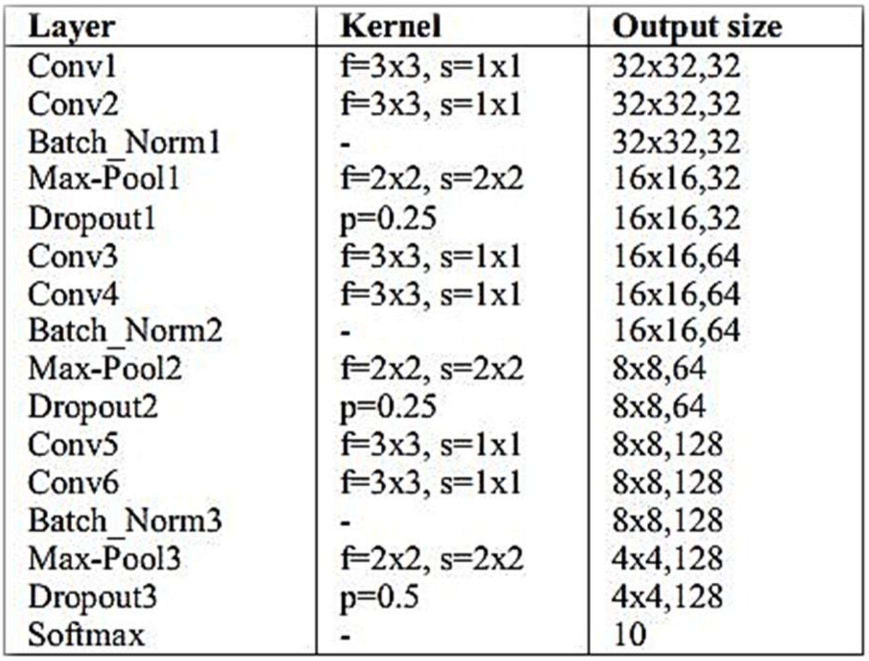
4.2 Purposed Neural Networks

This section provides the configuration details of the deep learning architectures utilized in this thesis, namely the conventional CNN (Section 4.2.1) and the more complex VGG 16 (Section 4.2.2)

4.2.1 CNN Architecture

We have used the CNN model from [77], here have used three repeated cells and each cell contains the two convolution layers, max pooling, batch normalization and dropout. The purposed architecture is relatively simple with total 6 convolution layers in which drop out and max-pooling are implemented after every two convolution layers. **Table 4.1** shows this technique. As it can be seen that each convolution layer consists of similar filter dimension of 3x3 and strides of 1x1, paddings are added to retain the size of input dimension. Dense layer is added only at the final output softmax layer. Regularization L2 is added on weights of all the convolution layers using the decay rate of 0.0001. In order get the faster convergence and to speed up the training time in less number of epochs batch normalization is also applied before every max-pooling[78].

Table 4.1: Proposed CNN Model



Layer	Kernel	Output size
Conv1	f=3x3, s=1x1	32x32,32
Conv2	f=3x3, s=1x1	32x32,32
Batch_Norm1	-	32x32,32
Max-Pool1	f=2x2, s=2x2	16x16,32
Dropout1	p=0.25	16x16,32
Conv3	f=3x3, s=1x1	16x16,64
Conv4	f=3x3, s=1x1	16x16,64
Batch_Norm2	-	16x16,64
Max-Pool2	f=2x2, s=2x2	8x8,64
Dropout2	p=0.25	8x8,64
Conv5	f=3x3, s=1x1	8x8,128
Conv6	f=3x3, s=1x1	8x8,128
Batch_Norm3	-	8x8,128
Max-Pool3	f=2x2, s=2x2	4x4,128
Dropout3	p=0.5	4x4,128
Softmax	-	10

- **Activation Function**

Softmax converts the logits, which is the output from the last linear layer of a neural network in a multi class classification, in to the probabilities. It takes the exponents of individual output and do normalization of each individual value by the addition of these exponents such that all probabilities output values should sums up to one. Softmax is normally added at the final layer of the neural network such that CNN, VGG-16 etc.

- **Optimizer**

We have used RMSProp [79], It is alike to gradient descent algorithm including momentum. The purpose of the RMSProp is to maintain the moving average of the gradient square and to divide the gradient with root of this average.

- **Loss Function**

As we are dealing with the multi class classification problem therefore we have used Categorical cross entropy loss in our case. Log loss, or Categorical cross entropy loss estimates the performance of the classification model by calculating loss, whose output is the probability estimation between 0 and 1. Loss increases when the predicated probability changes from the actual label.

$$-\sum_{c=1}^{M} y_{o,} clog(p_o, c) \tag{4.1}$$

Here, M depicts the number of classes, log is the natural log function, and y is binary indicator i.e. 0 or 1.

Table 4.2: Raw pixel accuracies for different datasets when conventional CNN is employed

Dataset	Raw Pixel Accuracy
MNIST	98.8%
CIFAR 10	87.5%
CIFAR 100	60.5%

4.2.2 VGG-16 Architecture

In literature, we have found that VGG-16 [66] [80] is the most robust neural network found for image classification problems. Therefore, we have employed this model in our proposed DCT based network. We have used ReLU [81] as an activation function and which is further followed by convolution layer and fully connected layer expect the last layer. We have used adaptive dropout only in fully connected layer. VGG-16 uses 3x3 filters in all its layers and 2x2 max pooling with stride 2 is used in this model.

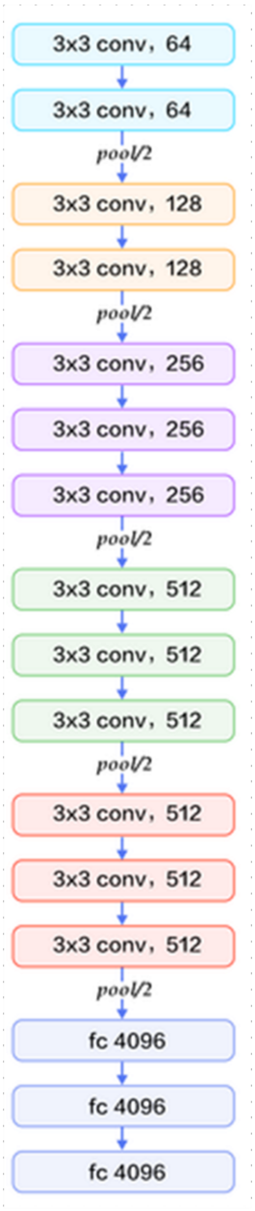


Figure 4.1: VGG-16 Architecture

Gradient stochastic descent SGD is used as optimizer. We have used Adaptive drop out and L2 regularization in order to minimize the over fitting [82].

Table 4.3: Raw pixel accuracies for different datasets when VGG-16 is employed

Dataset	**Raw Pixel Accuracy**
CIFAR 10	91.6%
CIFAR 100	70.3%

4.3 Classification from Important DCT Coefficients

In this section we have used purposed Deep Neural Network models for classification of important DCT coefficients extracted from each 8x8 and whole image. DCT coefficients containing the important information about the image are being extracted using the zigzag scanning method in both case. After selection of important coefficients inverse zigzag was performed and resultant matrix was given to Neural Network for classification purpose.

4.3.1 CNN Classification Results Using Whole Image as input to DCT

Table 4.4: MNIST Classification Results Calculation for DCT Based CNN Model

Num. of DCT coefficients	Maximum Validation Accuracy (%) MNIST
1024(100%)	98.69%
820(80.34%)	98.45%
520(50.7%)	98.23%
90(8.7%)	98.09%
80(7.8%)	97.66%
70(6.8%)	97.60%
60(5.8%)	97.50%
50(4.8%)	96.89%
40(3.9%)	96.30%

Table 4.4 shows the classification accuracy on MNIST dataset when using 32x32 as input to DCT network. It can be seen in the table that only by using 90 coefficients we can achieve around 98% accuracy in our purposed model. **Table 4.5** depicts the classification accuracy on CIFAR-10 and CIFAR-100 datasets.

Table 4.5: Results using Conventional CNN, DCT of the image taken as a whole

Percentage of DCT Coefficients utilized	Dataset	
	CIFAR 10	CIFAR 100
3072 (100%)	75.88%	48.83%
2700 (87%)	75.80%	48.68%
2100 (68%)	75.76%	48.42%
1920 (62.5%)	74.81%	47.68%
1536 (50%)	73.12%	46.32%

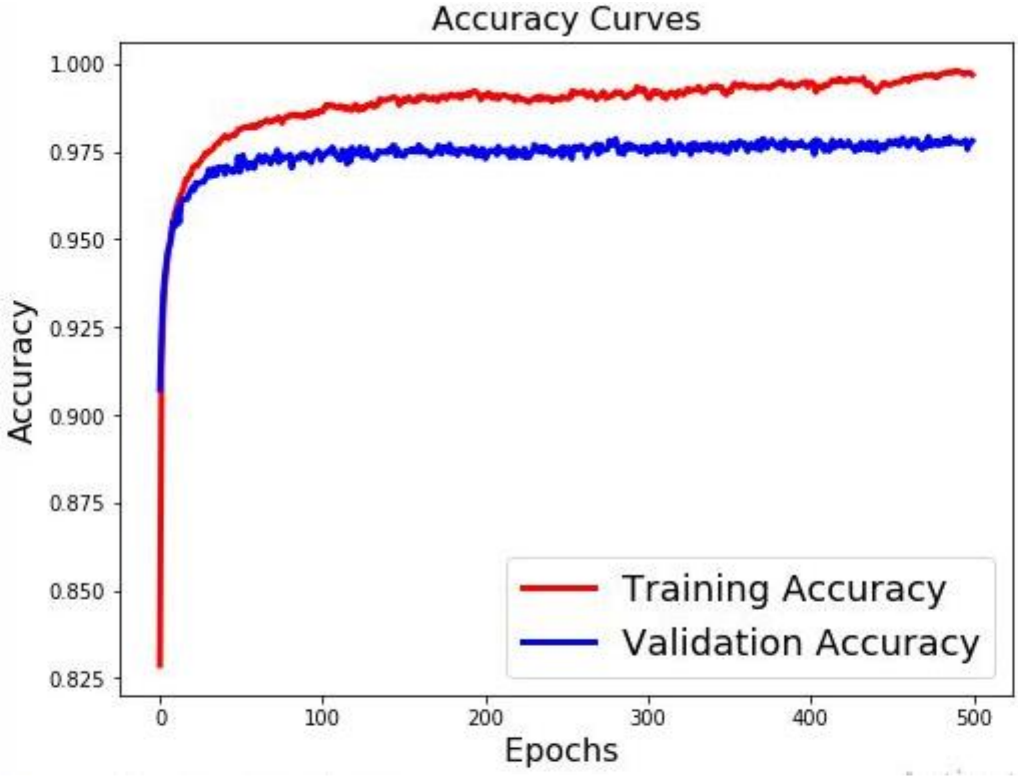


Figure 4.2: Classification Accuracy using only 90 DCT Coefficients in MNIST Dataset

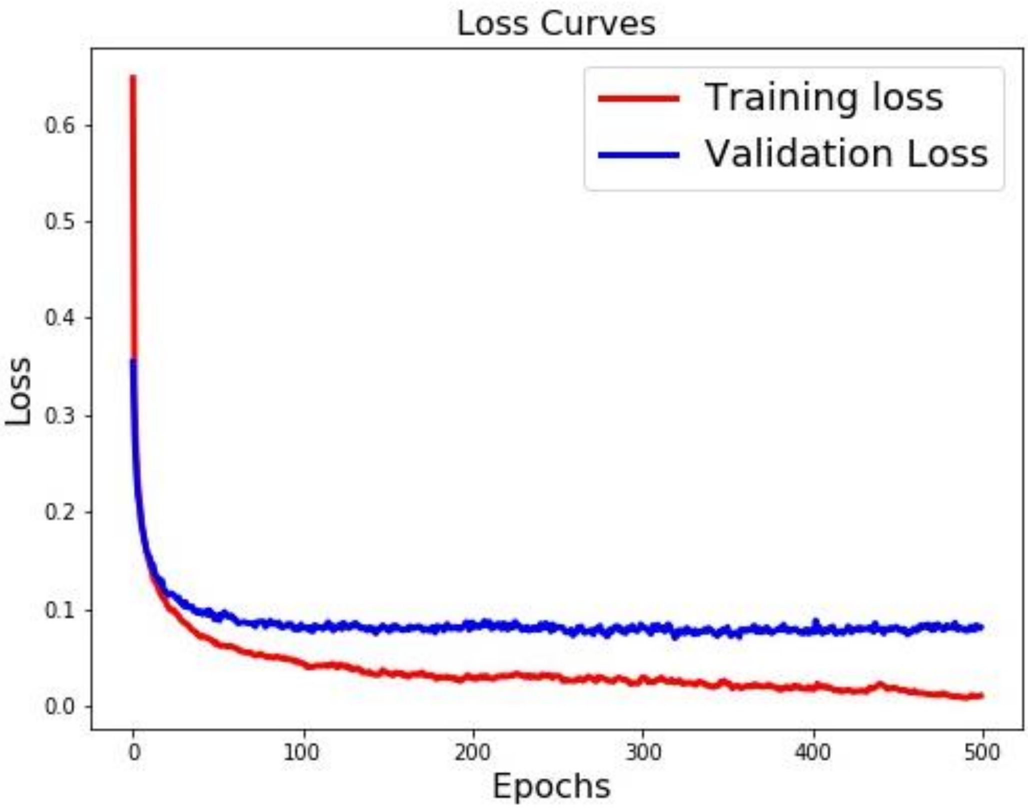


Figure 4.3: Loss Calculation by using only 90 DCT Coefficients in MNIST Dataset

4.3.2 VGG-16 Classification Results Using Whole Image as input to DCT

Table 4.6: Cifar-10 and Cifar-100 Classification Results Calculation for DCT Based VGG-16 Model (Whole Image)

Num. of DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
3072(100%)	80.74%	53.89%
3060	80.36%	53.54%
3000	80.54%	53.31%
2850	80.66%	53.24%
2700	80.19%	53.11%
2640	80.30%	53.23%
2460	80.28%	53.07%

2100	79.46%	53.02%
2007(65%)	78.50%	52.60%

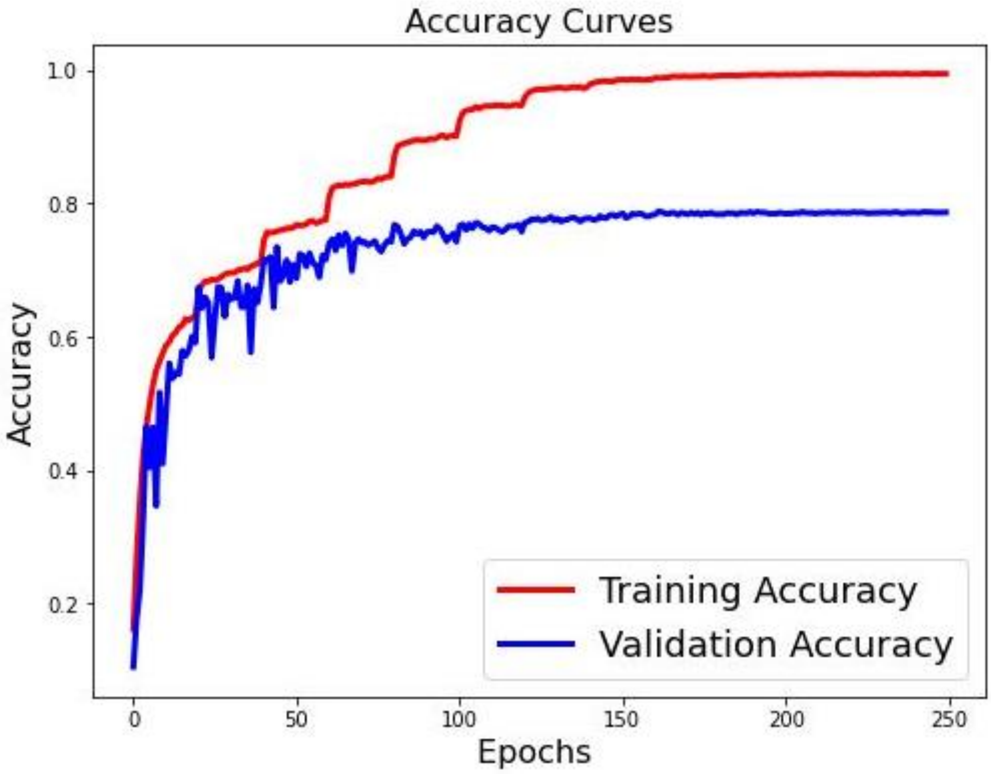


Figure 4.4: CIFAR-10 Classification Accuracy using 2007 DCT Coefficients from 3072

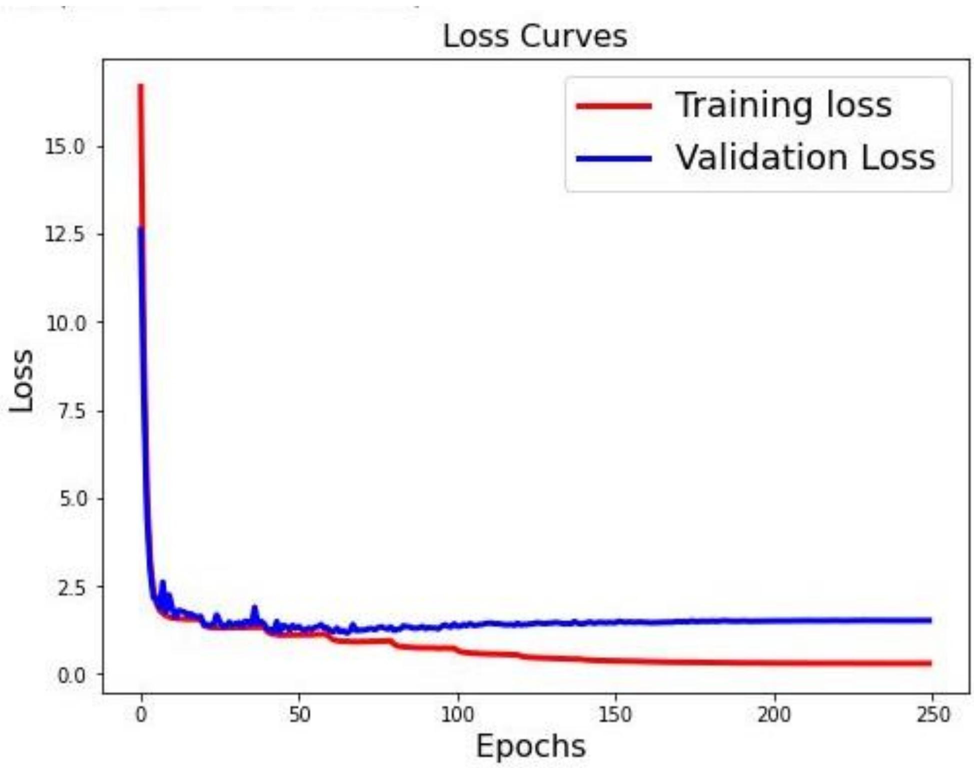


Figure 4.5: CIFAR-10 Loss Calculation using 2007 DCT Coefficients from 3072

4.3.3 CNN Classification Results Using 8x8 Block as input to DCT

Similar to section 4.3.1.1, here we have presented the classification results of all three MNIST, CIFAR-10 and CIFAR-100 datasets when using 8x8 block of image as input to DCT. We have performed the series of experiments on all three datasets. **Table 4.7** shows the classification results on MNIST dataset by taking 8x8 block of data as input and then doing classification on the most important DCT coefficients using zigzag and performing inverse zigzag on retaining DCT coefficients. **Table 4.8** presents the classification result of CIFAR-10 and CIFAR-100 datasets.

Table 4.7: MNIST Classification Results Calculation for DCT Based CNN Model (8x8 Input Size)

Num. of DCT coefficients	**Maximum Validation Accuracy (%) MNSIT**
64(100%)	98.78%
50	98.75%
40	98.62%
30	98.44%
20	98.30%
15(23%)	97.5%

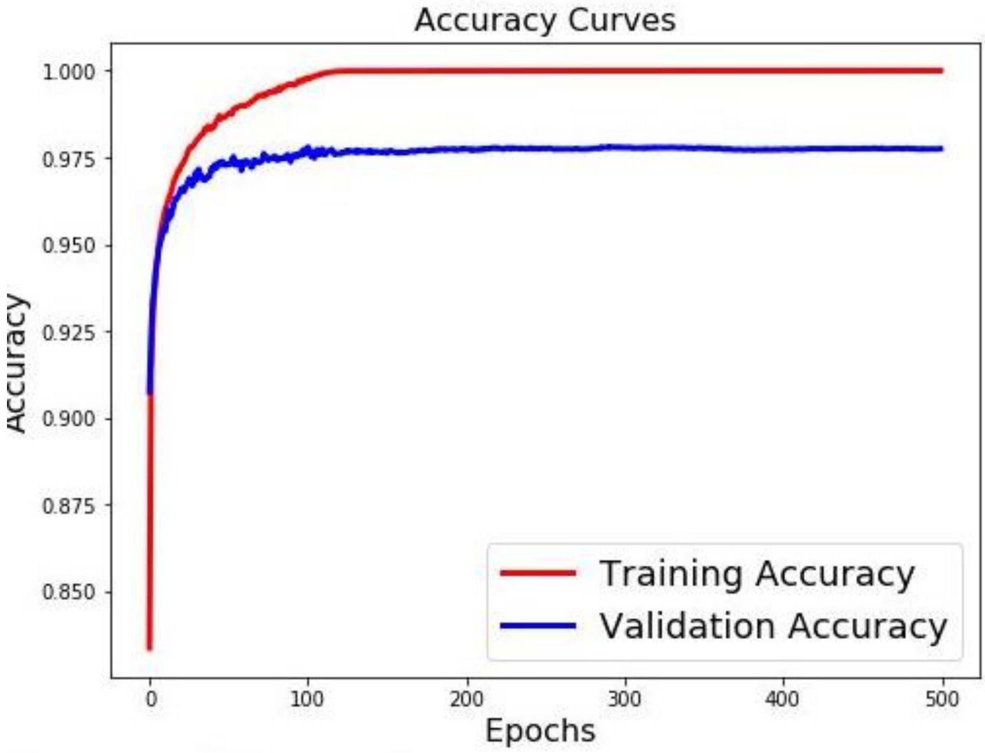


Figure 4.6: Classification Accuracy by using only 15 important DCT coefficients in MNIST Dataset

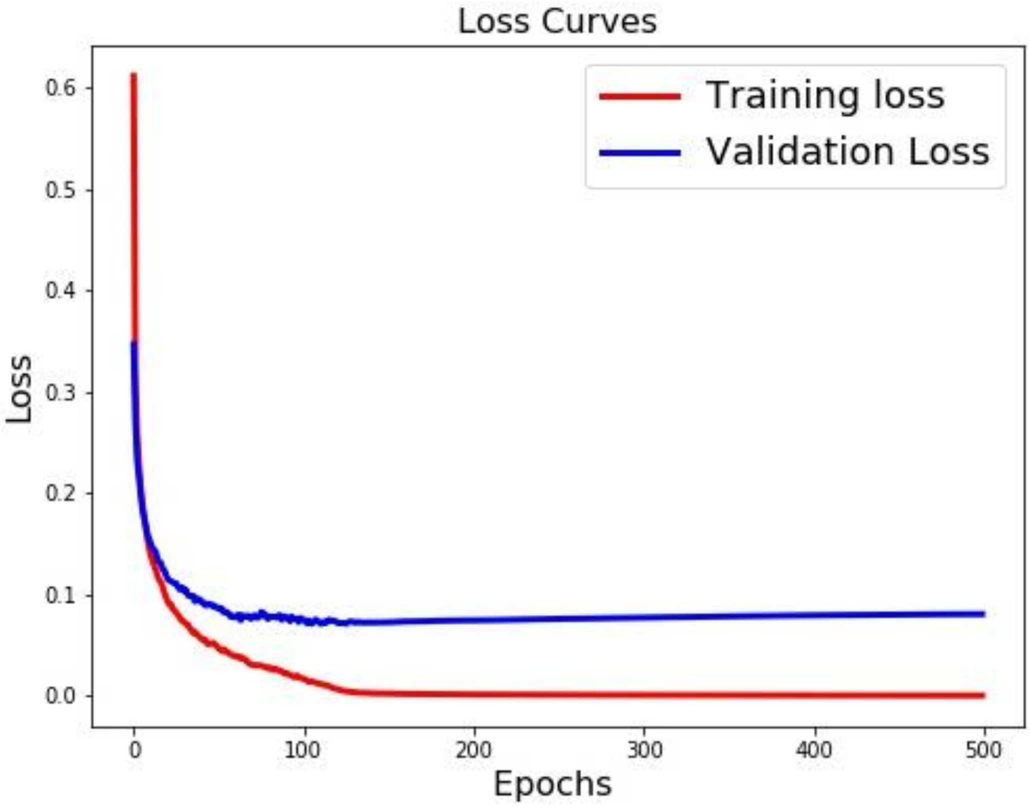


Figure 4.7: Loss Calculation by using only 15 important DCT coefficients in MNIST Dataset

Table 4.8: CIFAR-10 and CIFAR-100 Classification Results Calculation for DCT Based CNN Model (8x8 Input Size)

Num. of DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
64(100%)	80.05%	52.73%
50	80.48%	52.52%
40	78.07%	51.36%
30(46%)	78.13%	51.23%

In this section, we have used VGG-16 model for classification of important DCT coefficients extracted from each 8x8 and 32x32 (equal to image size) block of data. DCT coefficients containing the important information about the image are being extracted using the zigzag scanning method in both cases. After selection of important coefficients inverse zigzag was performed and resultant matrix was given to V-166 for classification purpose.

4.3.4 VGG-16 Classification Results Using 8x8 Block as input to DCT

Table 4.9: Cifar-10 and Cifar-100 Classification Results Calculation for DCT Based VGG-16 Model (8x8 Input Size)

Num. of selected DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
64(100%)	84.73%	60.07%
50	84.48%	60.04%
40	83.43%	59.59%
30(46%)	82.13%	59.03%

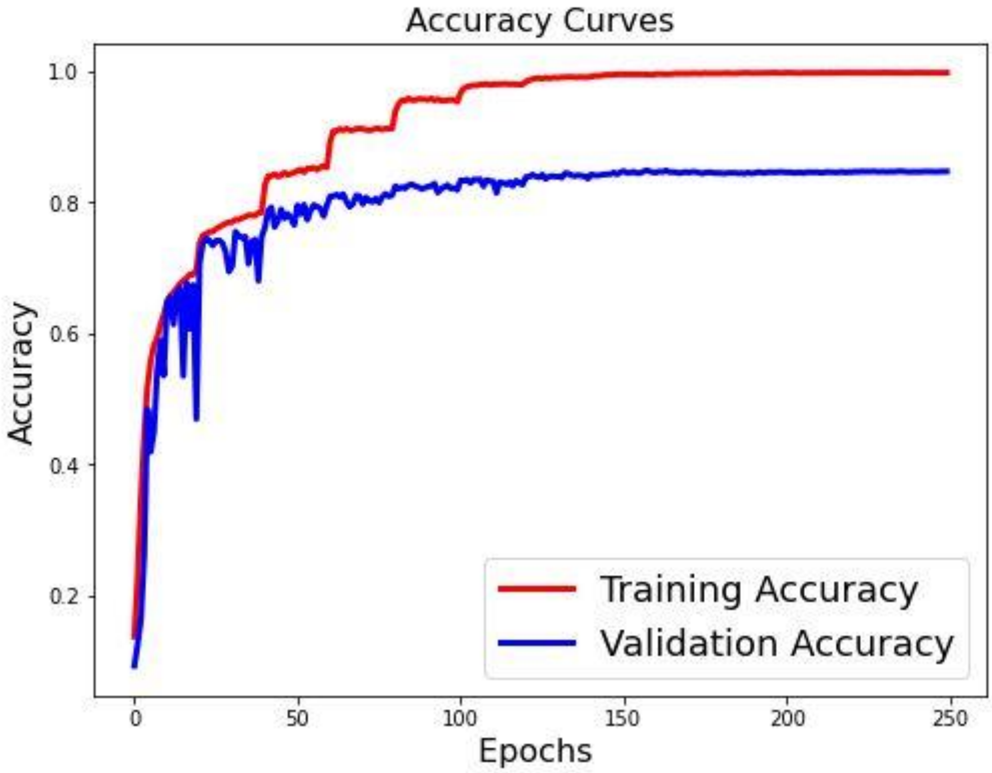


Figure 4.8: CIFAR-10 Classification Accuracy using 64 DCT Coefficients from each 8x8 Block

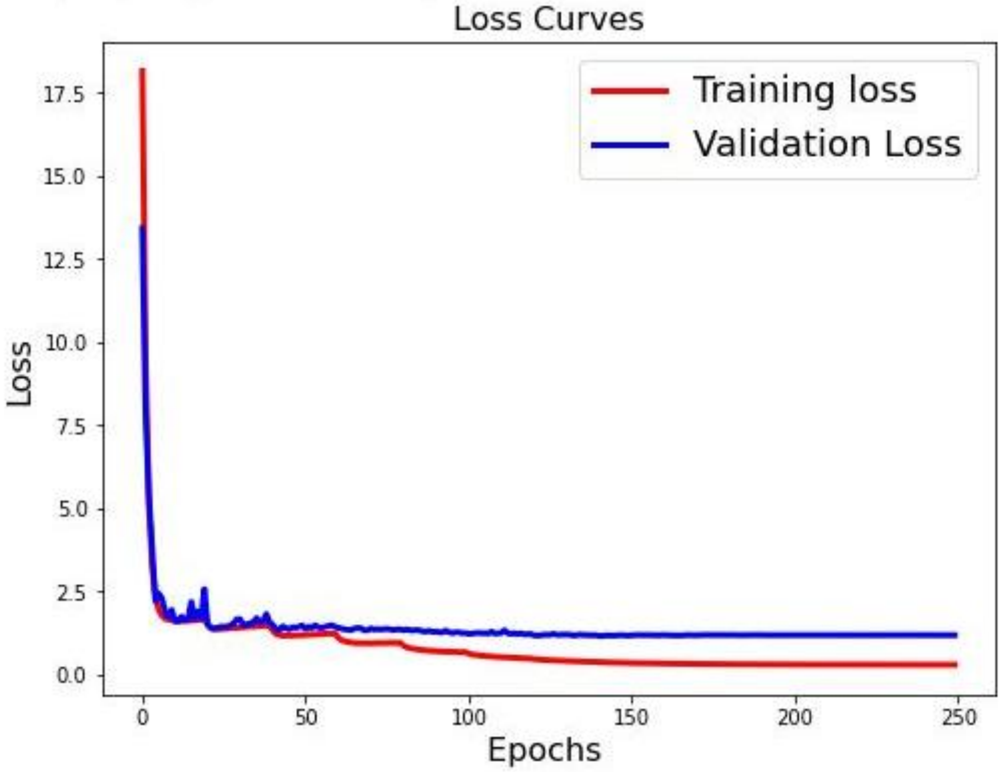


Figure 4.9: CIFAR-10 Loss Calculation using 64 DCT Coefficients from each 8x8 Block

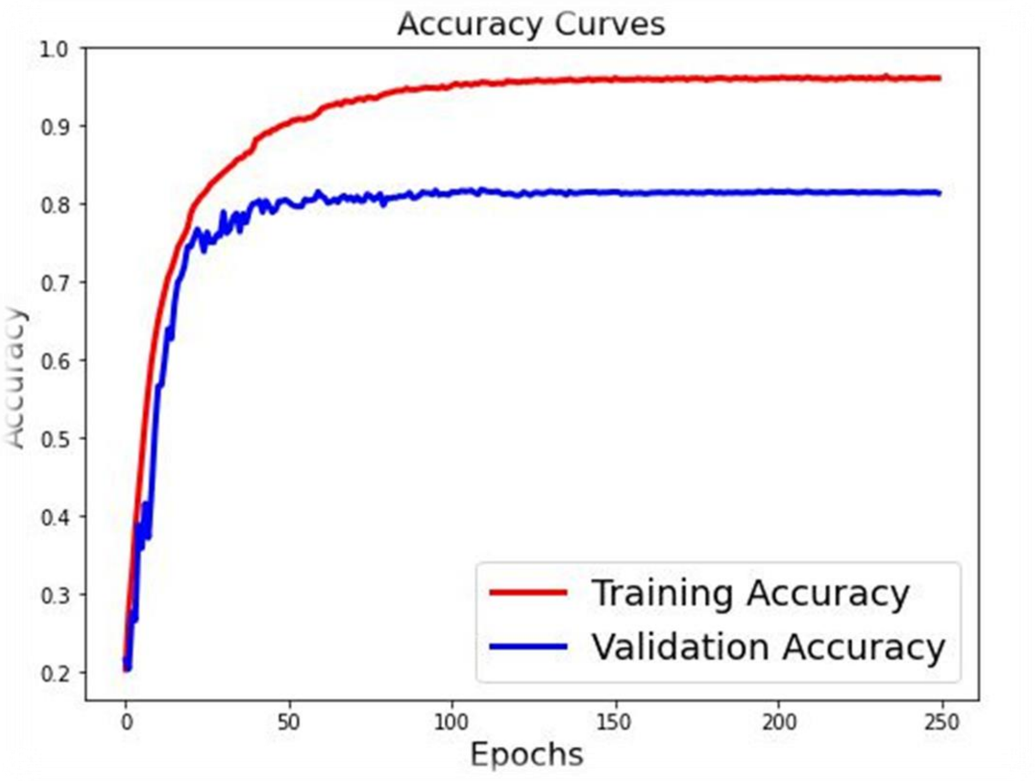


Figure 4.10: CIFAR-10 Classification Accuracy using 30 DCT Coefficients from each 8x8 Block

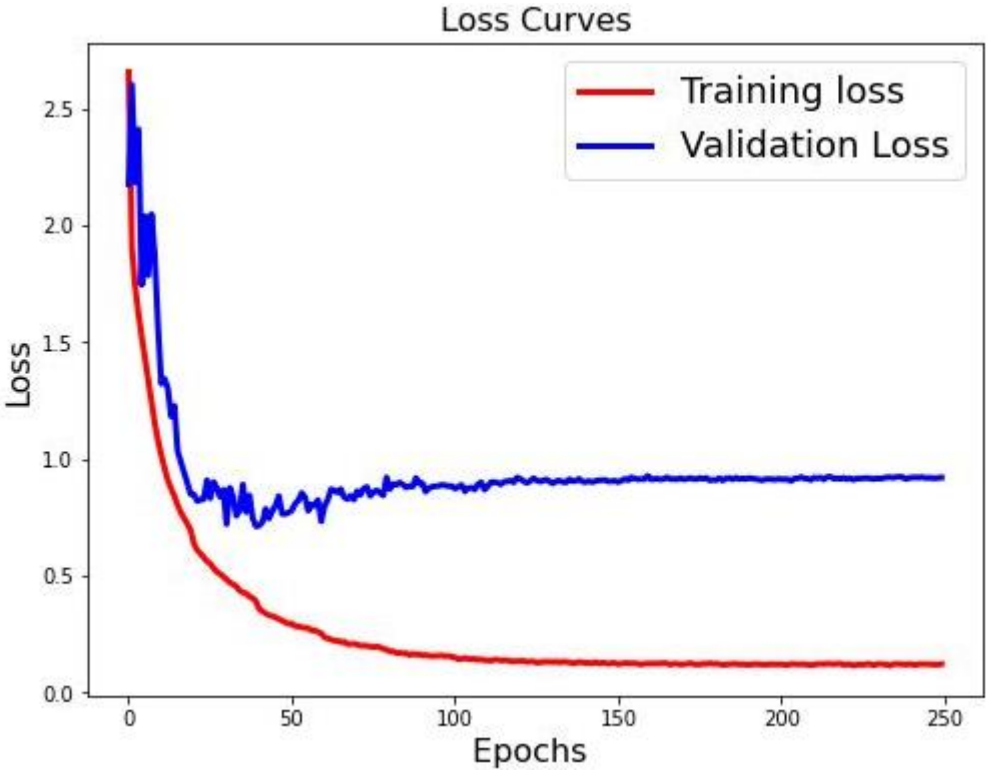


Figure 4.11: CIFAR-10 Loss Calculation using 30 DCT Coefficients from each 8x8 Block

4.4 Classification from Low Resolution Images

CNN and VGG-16 have been used as a classifiers to perform classification on the low resolution images which we have already compressed using by selecting the important DCT coefficient and discarding the least important coefficients. As discussed in the methodology part we have implemented this approach in both 8x8 block and also on whole image.

4.4.1 CNN Classification Results using Whole Image as input to DCT

Table 4.5 shows the classification accuracy on MNIST dataset when using 32x32 as input to DCT network. It can be seen in the table that only by using 50 coefficients we can achieve around 98% accuracy in our purposed model.

Table 4.10: MNIST Classification Results Calculation for DCT Based CNN Model (Whole Image)

Num. of DCT coefficients	Maximum Validation Accuracy (%) MNIST
1024(100%)	98.88%
90	98.74%
80	98.66%
70	98.54%
60	98.66%
50	97.89%
40	97.30%
30(3%)	95.28%

Following **Table 4.6** shows the classification accuracy on CIFAR-10 and CIFAR-100 datasets. As we can see from the results that instead of using all 3072 coefficients we can achieve the same accuracy of 87% with around 2000 important DCT coefficients.

Table 4.11: Cifar-10 and Cifar-100 Classification Results Calculation for DCT Based CNN Model (Whole Image)

Num. of DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
3072(100%)	87.74%	60.79%
3060	87.36%	60.54%
3000	87.54%	60.68%
2850	87.66%	60.34%
2700	87.19%	60.71%
2640	87.30%	60.12%
2460	87.28%	60.57%
2100	87.46%	60.40%
2007	86.50%	59.20%
1920(62%)	86.32%	59.15%

4.4.2 VGG-16 Classification Results using Whole Image as input to DCT

In this section VGG16 has been used as a classifier to perform classification on the low resolution images which we have already compressed using by selecting the important DCT coefficient and discarding the least important coefficients. As discussed in the methodology part we have implemented this approach in both 8x8 block and also on whole image.

Table 4.12: Cifar-10 and Cifar-100 Classification Results Calculation for DCT Based VGG-16 Model (Whole Image)

Num. of DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
3072(100%)	91.74%	70.79%
3060	91.36%	70.54%
3000	91.54%	70.68%
2850	91.66%	70.34%

2700	91.19%	70.71%
2640	91.30%	70.12%
2460	91.28%	70.57%
2100	91.46%	70.40%
2007	90.50%	69.20%
1920(62%)	90.32%	69.14%

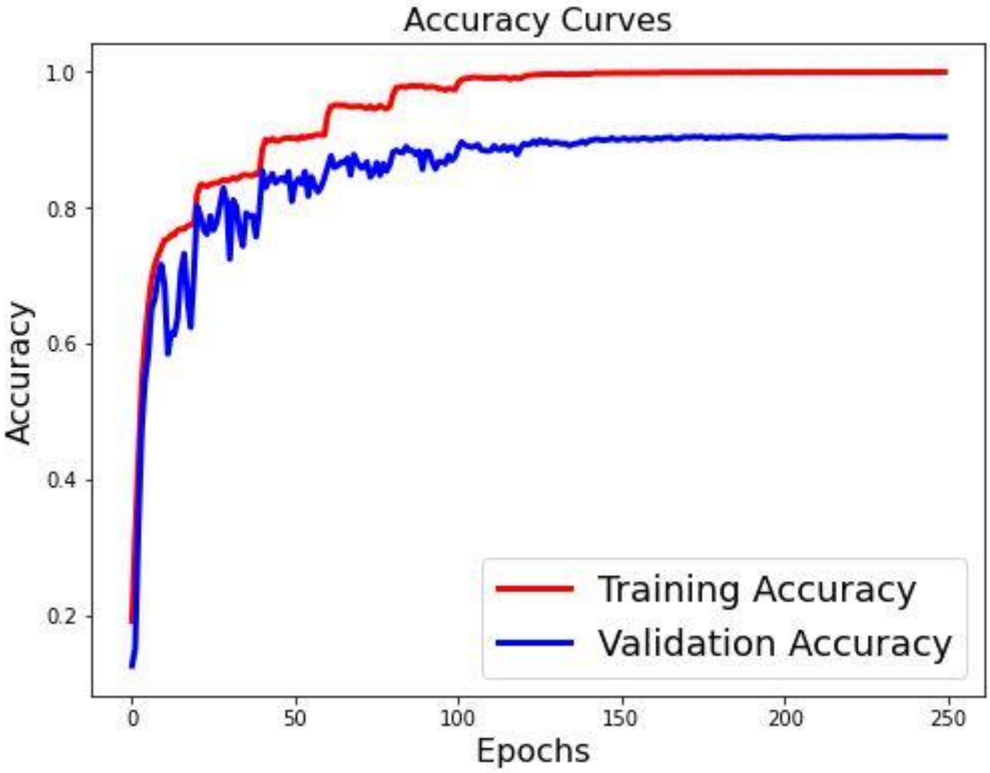


Figure 4.12: CIFAR-10 Classification Accuracy using 2100 DCT Coefficients from 3072

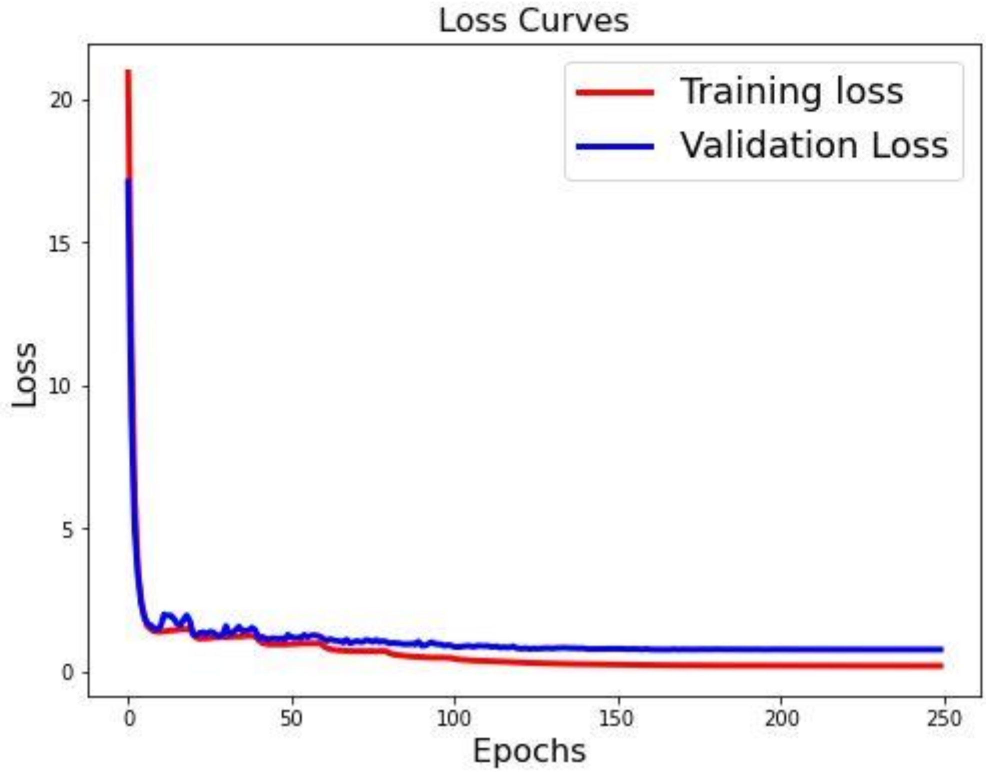


Figure 4.13: CIFAR-10 Loss Calculation using 2100 DCT Coefficients from 3072

4.4.3 CNN Classification Results Using 8x8 Block as input to DCT

In this section, we have presented the classification results of all three MNIST, CIFAR-10 and CIFAR-100 datasets when using 8x8 block of image as input to DCT. We have performed the series of experiments on all three datasets. **Table 4.3** shows the classification results on MNIST dataset by taking 8x8 block of data as input and then doing classification on low resolution images achieved after selecting the most important DCT coefficients using zigzag and performing the inverse DCT transform and inverse zigzag on retaining DCT coefficients.

Table 4.13: MNIST Classification Results and Loss Calculation for DCT Based CNN Model (8x8 Input Size)

Num. of DCT coefficients	Maximum Validation Accuracy (%)	Loss
64(100%)	98.7%	0.08%

50	98.6%	0.08%
40	98.4%	0.08%
30	98.3%	0.08%
20	97.9%	0.1%
15(23%)	97.8%	0.1%

Table 4.4 shows the classification accuracy on low resolution images in case of CIAFAR-10 and CIFAR-100 datasets. Same approach was used as mentioned for MNIST dataset except that each process was repeated for three times as CIFAR-10 and CIFAR-100 are RGB image datasets.

Table 4.14: Cifar-10 and Cifar-100 Classification Results Calculation for DCT Based CNN Model (8x8 Input Size)

Num. of DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
64(100%)	87.50%	60.32%
50	87.64%	60.04%
40	86.45%	59.59%
30	86.39%	59.35%
20(31%)	85.69%	59.46%

4.4.4 VGG-16 Classification Results by Using 8x8 Block as input to DCT

Table 4.15: Cifar-10 and Cifar-100 Classification Results Calculation for DCT Based VGG-16 Model (8x8 Input Size)

Num. of selected DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
64(100%)	91.73%	70.32%
50	91.64%	70.04%
40	90.45%	69.59%
30	90.09%	69.01%
20(31%)	89.69%	68.32%

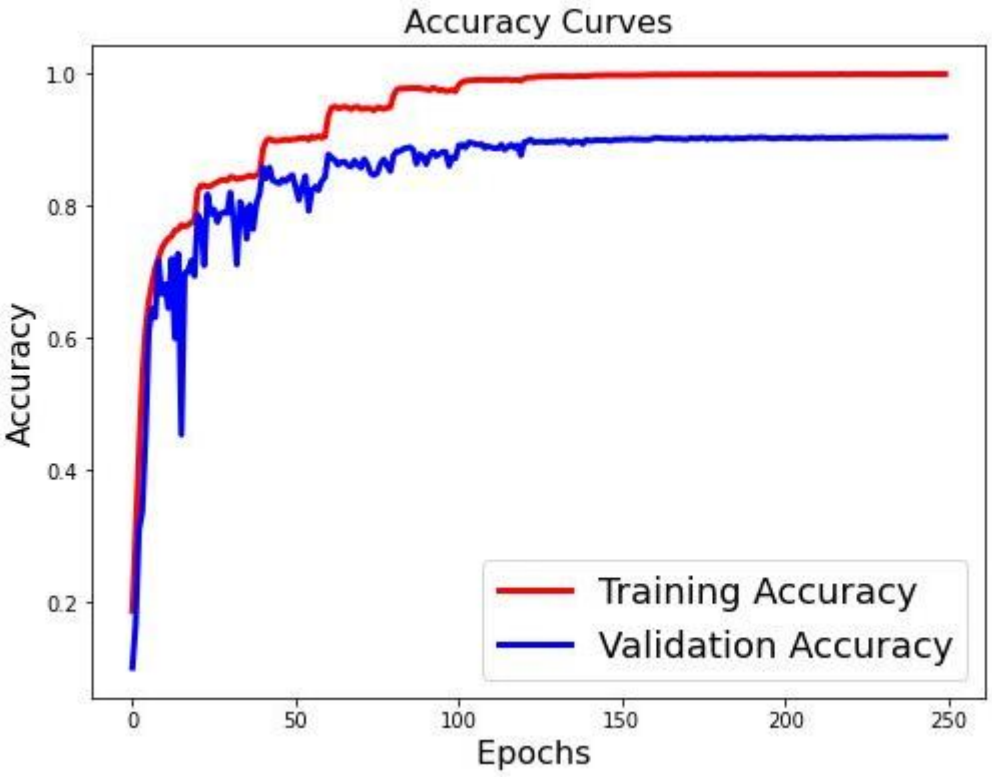


Figure 4.14: CIFAR-10 Classification Accuracy using 50 DCT Coefficients from each 8x8 Block

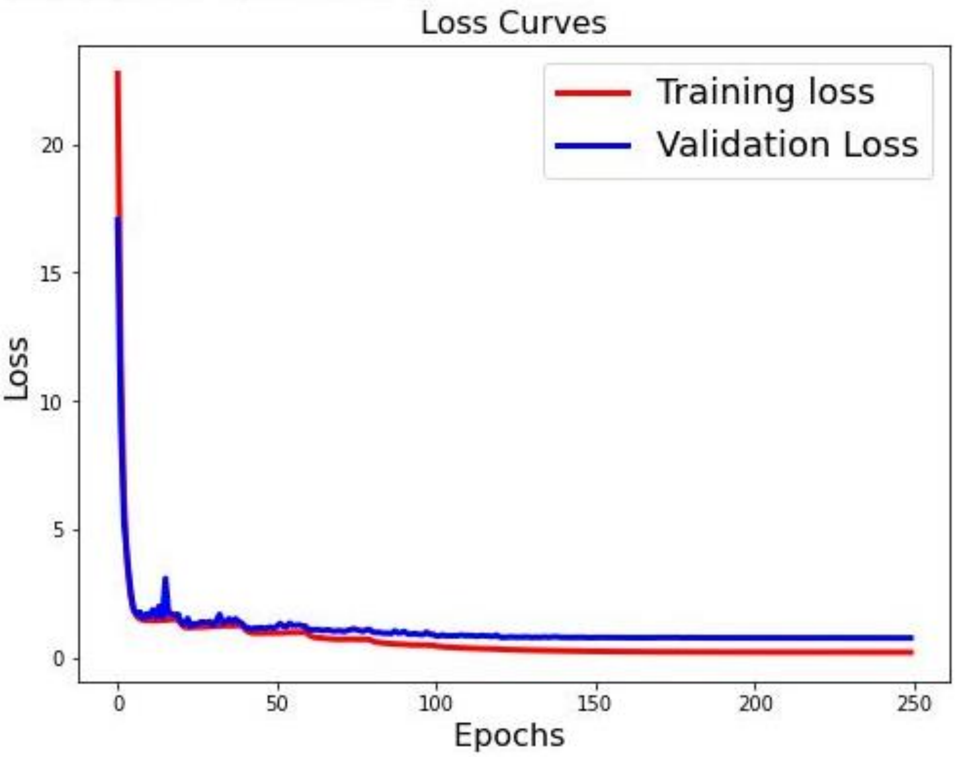


Figure 4.15: CIFAR-10 Loss Calculation using 50 DCT Coefficients from each 8x8 Block

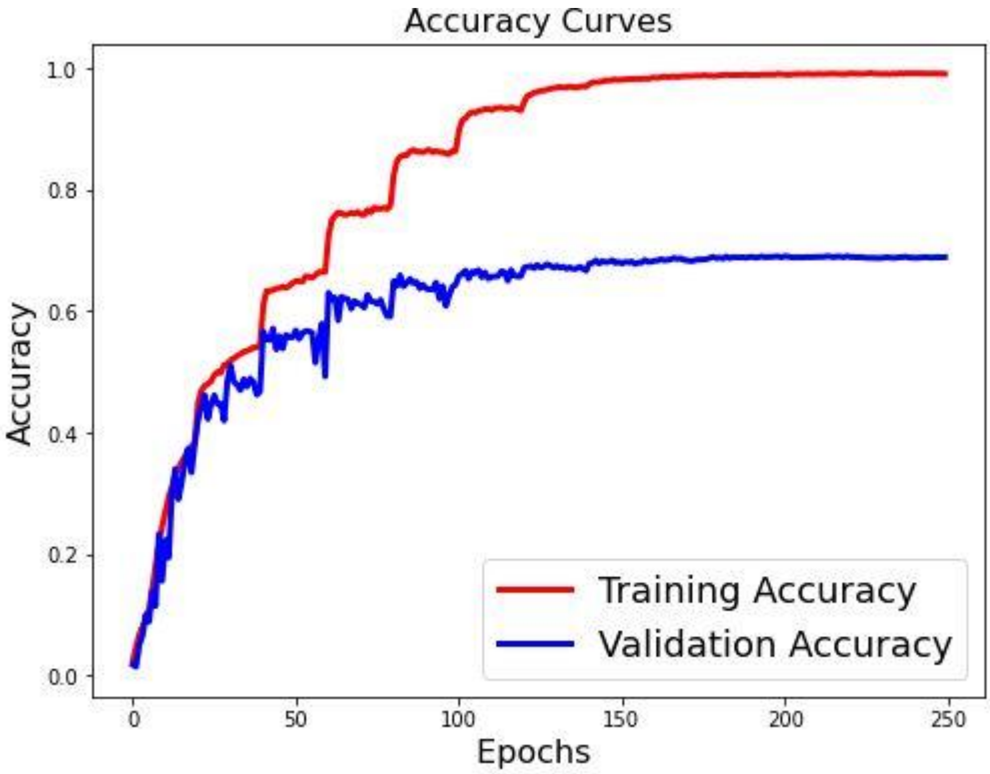


Figure 4.16: CIFAR-100 Classification Accuracy using 50 DCT Coefficients from each 8x8 Block

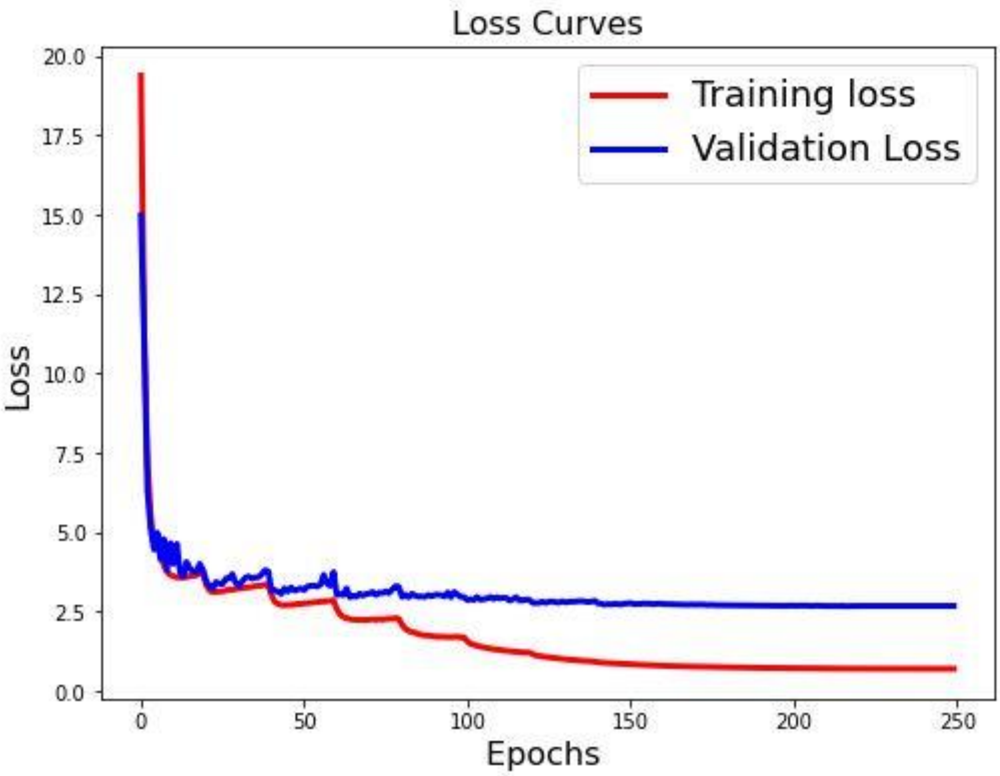


Figure 4.17: CIFAR-100 Loss Calculation using 50 DCT Coefficients from each 8x8 Block

4.5 Prediction Model Results

Classification accuracy of low resolution images containing DCT coefficients predicted from proposed Prediction Model is depicted in **Table 4.16**.

Table 4.16: Classification Accuracy on Low Resolution Images using VGG-16 (Input Size 8x8)

Num. of selected DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
64(100%)	91.50%	70.66%
50	90.84%	69.84%
40	90.45%	69.59%
30(31%)	90.39%	69.03%

Classification accuracy using DCT coefficients predicted from proposed prediction model is shown in **Table 4.17**

Table 4.17: Classification Accuracy on Predicted DCT Coefficients using VGG-16 (Input Size 8x8)

Num. of selected DCT coefficients	Maximum Validation Accuracy (%) CIFAR-10	Maximum Validation Accuracy (%) CIFAR-100
64(100%)	84.05%	60.3%
50	83.48%	60.5%
40	82.46%	59.3%
30(31%)	82.24%	58.8%

Chapter 5

Discussion and Limitations

CHAPTER 5: DISCUSSION AND LIMITATIONS

5.1 Discussion

This study presents the approaches to detect the important DCT coefficients from CIFAR-10 and CIFAR-100 RGB images. After extracting the important DCT coefficients needed to describe the image we have performed classification on remaining coefficients first by using low resolution images (after performing inverse discrete cosine transform on important DCT coefficients) and secondly by using those DCT as it is for classification tasks. Both approaches were tested initially by dividing each image in to equal 8x8 RGB blocks as an input to DCT module and then by using block size equal to image size (32x32) i.e. by taking DCT of whole image and discarding the redundant information (high frequency) information from each approach. A series of experiments have been performed using CNN and VGG-16 as classifiers and after performing a series of experiments on proposed CNN and VGG-16 we have found the promising accuracies using VGG-16 model. As it can be seen in following figures that as the number of DCT coefficients increases there is a slight increase in the recognition accuracy. As it can be seen in **Figure 5.1**, the best accuracy for CIFAR-10 achieved is 91% with only 40 DCT coefficients extracted from 64 DCT vector and same 91% can be achieved in with using only 2100 important DCT efficient extracted from each 3072. This is the same accuracy which we achieve after classification of raw pixel data of CIFAR-10.

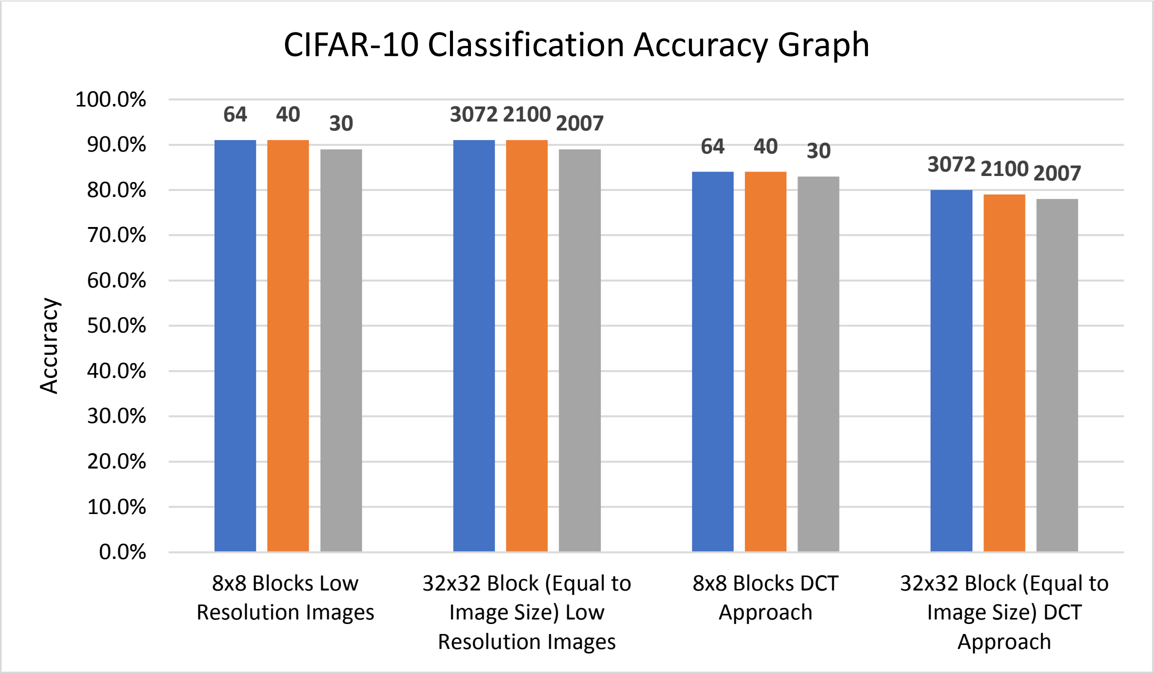


Figure 5.1: CIFAR-10 Classification Accuracy using VGG-16 for different number of DCT coefficients retained

As depicted in **Figure 5.2**, best accuracy achieved in case of CIFAR-100 is around 70% by using 40 important DCT coefficients from each 64 DCT vector and same accuracy is achieved by we take only 2100 significant DCT coefficients out of 3072 that is the same as raw pixels.

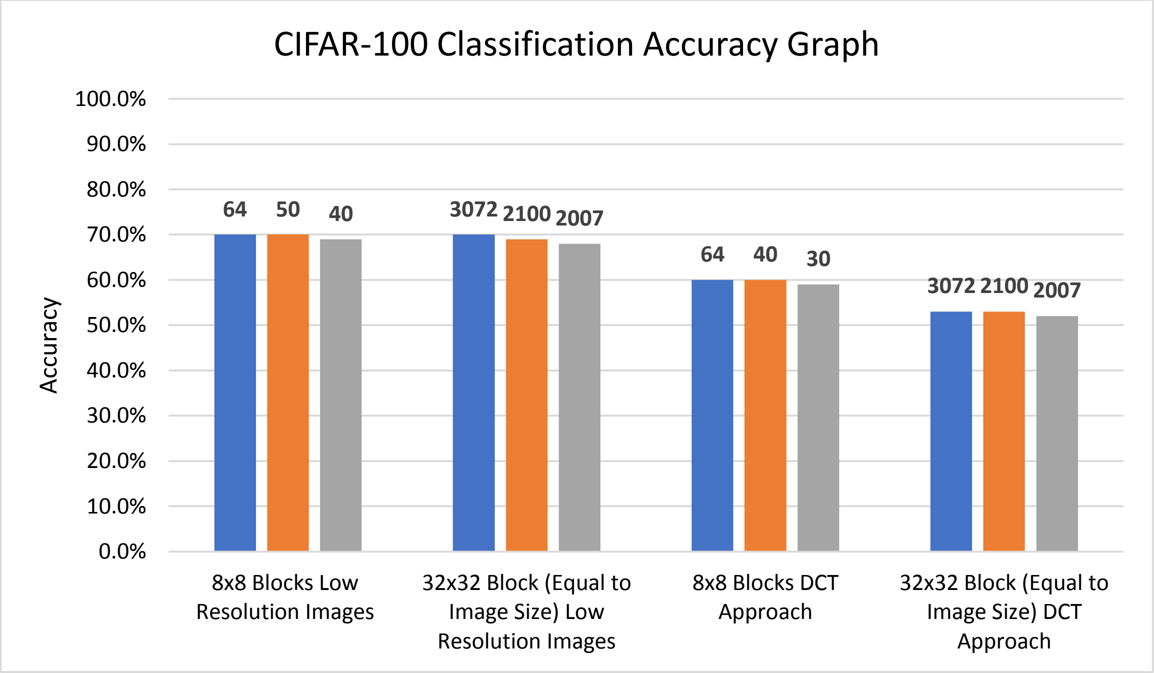


Figure 5.2: CIFAR-100 Classification Accuracy using VGG-16 for different number of DCT coefficients retained

5.2 Limitations

As Convolution Neural Networks are majorly used for image classification and only accept the 4D array as input (batch size, height, width, depth). Therefore, in order to get the original size of the image we had to perform the zero padding in place of discarded DCT coefficients which is limitation to reduce the training time.

Chapter 4

Conclusion and Future Work

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we have purposed an approach for implementing the convolution neural networks to learn from the frequency representation of data with motivation to reduce the bandwidth and storage cost required to transfer the large visual size data. By implementing this approach, we have contributed in the edge computing domain whose works on increasing the speed of data processing which in turns helps in lowering the dependency on cloud. For this purpose, a comprehensive study of literature was performed in order to identify the different image compression approaches and the use of convolution neural networks in the field of image compression tasks. After the detailed literature review, we have taken the advantage of Discrete Cosine Transform coefficients which is widely used in JPEG compression techniques. We have performed a series of experiments by taking the significant DCT coefficients. After achieving the important DCT coefficients from images we have also applied the classification models on the compact representation of Grey scale and RGB datasets (MNIST, CIFAR-10 and CIFAR-100). We have used VGG-16 and purposed CNN architecture for classification purpose. Our approach depicts that after discarding the redundant frequency components present in the bottom right of each DCT transformed block and by utilizing the low level features in DCT domain we can achieve almost the same accuracy which we were attaining from raw pixel data. The evolution of the results showed by using only the important DCT coefficients, an accuracy of the around 98% in MNIST, 91% in CIFAR-10 and 70% in CIFAR-100 is achieved on the classification problem which is the same as raw pixel data.

6.2 Future Work

Future work includes the more generalization of these neural networks to support the more image datasets. Also it will be focused on extending the proposed framework to more deep learning networks. In addition to that, compression of video data can also be achieved by extending these deep neural networks to video processing domain.

REFERENCES

1. Cao, K., et al., *An Overview on Edge Computing Research.* IEEE Access, 2020. **8**: p. 85714-85728.
2. Maswood, M.M.S., et al., *A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in a Cooperative Three-Layer Fog-Cloud Computing Environment.* IEEE Access, 2020. **8**: p. 113737-113750.
3. Shahzadi, S., et al., *Multi-access edge computing: open issues, challenges and future perspectives.* Journal of Cloud Computing, 2017. **6**(1): p. 30.
4. Shi, W., et al., *Edge computing: Vision and challenges.* IEEE internet of things journal, 2016. **3**(5): p. 637-646.
5. Ahmed, N., T. Natarajan, and K.R. Rao, *Discrete cosine transform.* IEEE transactions on Computers, 1974. **100**(1): p. 90-93.
6. Qiu, H., et al., *Deep Residual Learning based Enhanced JPEG Compression in the Internet of Things.* IEEE Transactions on Industrial Informatics, 2020.
7. Kitakami, M. and K. Tai. *Lossless Image Compression by PPM-Based Prediction Coding*. in *2009 Data Compression Conference*. 2009. IEEE.
8. Sathappan, S. and P.S. Babu. *A new model for color image compression using modified hierarchical prediction*. in *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. 2015. IEEE.
9. Nashat, A.A. and N.H. Hassan. *Image compression based upon wavelet transform and a statistical threshold*. in *2016 International Conference on Optoelectronics and Image Processing (ICOIP)*. 2016. IEEE.
10. Naidu, B.R. and M.P. Babu. *A novel framework for JPEG image compression using baseline coding with parallel process*. in *2014 IEEE International Conference on Computational Intelligence and Computing Research*. 2014. IEEE.
11. Jha, M., A.S. Yumnam, and D. Raju. *Comparison between image codecs: JPEG and JPEG2000*. in *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*. 2014. IEEE.
12. Qu, Z., W. Luo, and J. Huang, *A framework for identifying shifted double JPEG compression artifacts with application to non-intrusive digital image forensics.* Science China Information Sciences, 2014. **57**(2): p. 1-18.

13. Pang, C.-Y., et al., *Signal and image compression using quantum discrete cosine transform.* Information Sciences, 2019. **473**: p. 121-141.

14. Kurniawan, A., T.W. Purboyo, and A.L. Prasasti, *Implementation of image compression using discrete cosine transform (dct) and discrete wavelet transform (dwt).* International Journal of Applied Engineering Research, 2017. **12**(23): p. 13951-13958.

15. Wang, M., B. Liu, and H. Foroosh. *Factorized convolutional neural networks*. in *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017.

16. Li, T., et al. *Compressing convolutional neural networks via factorized convolutional filters*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

17. Song, D., P. Zhang, and F. Li. *Speeding Up Deep Convolutional Neural Networks Based on Tucker-CP Decomposition*. in *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*. 2020.

18. Ghosh, T. *Towards a new interpretation of separable convolutions*. in *2017 Intelligent Systems Conference (IntelliSys)*. 2017. IEEE.

19. Mukherjee, D. and S. Mukhopadhyay, *Fast hardware architecture for 2-d separable convolution operations.* IEEE Transactions on Circuits and Systems II: Express Briefs, 2018. **65**(12): p. 2042-2046.

20. Teng, C.-F., et al. *Low-complexity recurrent neural network-based polar decoder with weight quantization mechanism*. in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019. IEEE.

21. Liu, B., et al. *Sparse convolutional neural networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

22. Molchanov, P., et al. *Importance estimation for neural network pruning*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

23. SAPNA, B. and K. SUDHA. *Secured Transmission with Enhanced Security for Grayscale Images using Visual Cryptography*. in *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*. 2018. IEEE.

24. Hurtik, P. and I. Perfilieva. *A hybrid image compression algorithm based on jpeg and fuzzy transform*. in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2017. IEEE.

25. Youniss, R.Y.B. and A. Awajan. *An adaptive JPEG compression-encryption system using optimized local quantization matrices*. in *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*. 2017. IEEE.

26. ARTUĞER, F. and F. ÖZKAYNAK. *Fractal image compression method for lossy data compression*. in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. 2018. IEEE.

27. Wallace, G.K., *The JPEG still picture compression standard.* IEEE transactions on consumer electronics, 1992. **38**(1): p. xviii-xxxiv.

28. Chen, W.-H. and W. Pratt, *Scene adaptive coder.* IEEE Transactions on Communications, 1984. **32**(3): p. 225-232.

29. Zhang, J., et al., *A Deep Learning Approach in the Discrete Cosine Transform Domain to Median Filtering Forensics.* IEEE Signal Processing Letters, 2020. **27**: p. 276-280.

30. Rajesh, B., M. Javed, and S. Srivastava. *DCT-CompCNN: A Novel Image Classification Network Using JPEG Compressed DCT Coefficients*. in *2019 IEEE Conference on Information and Communication Technology*. 2019. IEEE.

31. Borhanuddin, B., et al. *Small-Scale Deep Network for DCT-Based Images Classification*. in *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*. 2019. IEEE.

32. Hossain, M.T., et al. *Distortion robust image classification using deep convolutional neural network with discrete cosine transform*. in *2019 IEEE International Conference on Image Processing (ICIP)*. 2019. IEEE.

33. Sun, M., et al., *Reduction of JPEG compression artifacts based on DCT coefficients prediction.* Neurocomputing, 2020. **384**: p. 335-345.

34. Zou, X., et al. *High speed deep networks based on discrete cosine transformation*. in *2014 IEEE International Conference on Image Processing (ICIP)*. 2014. IEEE.

35. Liu, Z., et al. *DeepN-JPEG: A deep neural network favorable JPEG-based image compression framework.* in *Proceedings of the 55th Annual Design Automation Conference*. 2018.

36. Gueguen, L., et al. *Faster neural networks straight from jpeg*. in *Advances in Neural Information Processing Systems*. 2018.

37. Ulicny, M. and R. Dahyot. *On using cnn with dct based image data*. in *Proceedings of the 19th Irish Machine Vision and Image Processing conference IMVIP*. 2017.

38. Messaoudi, A., F. Benchabane, and K. Srairi, *DCT-based color image compression algorithm using adaptive block scanning.* Signal, Image and Video Processing, 2019. **13**(7): p. 1441-1449.

39. Deguerre, B., C. Chatelain, and G. Gasso. *Fast object detection in compressed JPEG Images*. in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019. IEEE.

40. Ghosh, A. and R. Chellappa. *Deep feature extraction in the dct domain*. in *2016 23rd International Conference on Pattern Recognition (ICPR)*. 2016. IEEE.

41. Ng, K. and L. Cheng. *Artificial neural network for discrete cosine transform and image compression*. in *Proceedings of the fourth international conference on Document Analysis and Recognition*. 1997. IEEE.

42. Çalik, R.C. and M.F. Demirci. *Cifar-10 image classification with convolutional neural networks for embedded systems*. in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*. 2018. IEEE.

43. Krizhevsky, A., I. Sutskever, and G.E. Hinton. *Imagenet classification with deep convolutional neural networks*. in *Advances in neural information processing systems*. 2012.

44. Doon, R., T.K. Rawat, and S. Gautam. *Cifar-10 Classification using Deep Convolutional Neural Network*. in *2018 IEEE Punecon*. IEEE.

45. Egmont-Petersen, M., D. de Ridder, and H. Handels, *Image processing with neural networks—a review.* Pattern recognition, 2002. **35**(10): p. 2279-2301.

46. Dony, R.D. and S. Haykin, *Neural network approaches to image compression.* Proceedings of the IEEE, 1995. **83**(2): p. 288-303.

47. Daugman, J.G., *Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression.* IEEE Transactions on acoustics, speech, and signal processing, 1988. **36**(7): p. 1169-1179.

48. Jiang, J., *Image compression with neural networks–a survey.* Signal processing: image Communication, 1999. **14**(9): p. 737-760.

49. Luttrell, S. *Image compression using a neural network.* in *International Geoscience and Remote Sensing Symposium,'Remote Sensing: Moving Toward the 21st Century'.* 1988. IEEE.

50. Ciregan, D., U. Meier, and J. Schmidhuber. *Multi-column deep neural networks for image classification*. in *2012 IEEE conference on computer vision and pattern recognition*. 2012. IEEE.

51. Du, C. and S. Gao, *Image segmentation-based multi-focus image fusion through multi-scale convolutional neural network.* IEEE access, 2017. **5**: p. 15750-15761.

52. Ji, S., et al., *3D convolutional neural networks for human action recognition.* IEEE transactions on pattern analysis and machine intelligence, 2012. **35**(1): p. 221-231.

53. Yan, C., et al., *A fast uyghur text detector for complex background images.* IEEE Transactions on Multimedia, 2018. **20**(12): p. 3389-3398.

54. Yan, C., et al., *Cross-modality bridging and knowledge transferring for image understanding.* IEEE Transactions on Multimedia, 2019. **21**(10): p. 2675-2685.

55. Yan, C., et al., *Stat: spatial-temporal attention mechanism for video captioning.* IEEE transactions on multimedia, 2019. **22**(1): p. 229-241.

56. Chuang, M.-C., J.-N. Hwang, and K. Williams, *A feature learning and object recognition framework for underwater fish images.* IEEE Transactions on Image Processing, 2016. **25**(4): p. 1862-1872.

57. Chuang, M.-C., J.-N. Hwang, and K. Williams. *Supervised and unsupervised feature extraction methods for underwater fish species recognition*. in *2014 ICPR Workshop on Computer Vision for Analysis of Underwater Imagery*. 2014. IEEE.

58. Makhoul, J., *A fast cosine transform in one and two dimensions.* IEEE Transactions on Acoustics, Speech, and Signal Processing, 1980. **28**(1): p. 27-34.

59. Narasimha, M. and A. Peterson, *On the computation of the discrete cosine transform.* IEEE Transactions on Communications, 1978. **26**(6): p. 934-936.

60. Er, M.J., W. Chen, and S. Wu, *High-speed face recognition based on discrete cosine transform and RBF neural networks.* IEEE Transactions on neural networks, 2005. **16**(3): p. 679-691.

61. Birvinskas, D., et al., *Fast DCT algorithms for EEG data compression in embedded systems.* Computer Science and Information Systems, 2015. **12**(1): p. 49-62.

62. Sridhar, D. and I.M. Krishna. *Brain tumor classification using discrete cosine transform and probabilistic neural network*. in *2013 International Conference on Signal Processing, Image Processing & Pattern Recognition*. 2013. IEEE.

63. Russakovsky, O., et al., *Imagenet large scale visual recognition challenge.* International journal of computer vision, 2015. **115**(3): p. 211-252.

64. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

65. Zeiler, M.D. and R. Fergus. *Visualizing and understanding convolutional networks*. in *European conference on computer vision*. 2014. Springer.

66. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.

67. Szegedy, C., et al. *Going deeper with convolutions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

68. Dodge, S. and L. Karam. *Understanding how image quality affects deep neural networks*. in *2016 eighth international conference on quality of multimedia experience (QoMEX)*. 2016. IEEE.

69. Li, Y. and Y. Yuan. *Convergence analysis of two-layer neural networks with relu activation*. in *Advances in neural information processing systems*. 2017.

70. Karlik, B. and A.V. Olgac, *Performance analysis of various activation functions in generalized MLP architectures of neural networks.* International Journal of Artificial Intelligence and Expert Systems, 2011. **1**(4): p. 111-122.

71. Zhang, Z. *Improved adam optimizer for deep neural networks*. in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. 2018. IEEE.

72. De Myttenaere, A., et al., *Mean absolute percentage error for regression models.* Neurocomputing, 2016. **192**: p. 38-48.

73. Abadi, M., et al., *Tensorflow: Large-scale machine learning on heterogeneous distributed systems.* arXiv preprint arXiv:1603.04467, 2016.

74. Chollet, F., *Keras: The python deep learning library.* ascl, 2018: p. ascl: 1806.022.

75. LeCun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.

76. Krizhevsky, A. and G. Hinton, *Learning multiple layers of features from tiny images.* 2009.

77. Kumar, A., *Achieving 90% accuracy in Object Recognition Task on CIFAR-10 Dataset with Keras: Convolutional Neural Networks – Machine Learning in Action,.* 2018 [online]. **[Accessed: 05-Jun-2020].**

78. Ioffe, S. and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* arXiv preprint arXiv:1502.03167, 2015.

79. Tieleman, T. and G. Hinton, *Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning.* Technical Report., 2017.