M.Tech (VLSI and Embedded Design) Akansha Lal 2023

Accurate Image Classification for Improved Vision Recognition using Hierarchical Transfer Learning

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

VLSI Design and Embedded System

Submitted By:

AKANSHA LAL

2K21/VLS/03

Under the supervision of

Dr A. K. Singh, Professor (ECE)

&

Mr Vinay Kumar, Assistant Professor (ECE)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

May, 2023

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Akansha Lal, 2K21/VLS/03 student of MTech (VLSI Design and Embedded System), hereby declare that the project dissertation titled "**Accurate Image Classification for Improved Vision Recognition using Hierarchical Transfer Learning**" which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associate ship, Fellowship or other similar title or recognition.

Place: Delhi — Akansha Lal

Date: — 2K21/VLS/03

ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Accurate Image Classification for Improved Vision Recognition using Hierarchical Transfer Learning**" which is submitted by Akansha Lal, 2K21/VLS/03 Computer science and engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this university or elsewhere.

Place: Delhi

Date:

Mr. Vinay Kumar
Supervisor

Dr. Alok Kumar Singh
Supervisor

ABSTRACT

Image classification is an important task by computer vision with applications like object recognition to medical diagnosis. In last few years, transfer learning has emerged as effective technique to improve the image classification performance by utilizing pre-trained models knowledge on large scale.

The research proposes a novel method for image classification using transfer learning to harness the hierarchical relations classes and enhance the accuracy of vision recognition system.

A most suitable dataset is collected and then preprocessed, including steps such as normalization, resizing, data cleaning and argumentation. The design and implementation is done using hierarchical transfer learning framework. The trained model is evaluated on performance metrics like precision, accuracy, loss compared against baseline model to measure the improvement achieved.

The Python programming language, along with the TensorFlow framework, and Google Colaboratory hardware, was utilized for this thesis. Existing models available online were selected and modified accordingly.

The proposed methodology harnesses the hierarchical structure, resulting in enhanced accuracy for image classification tasks. The research approach presented in this study contributes to the progress of vision recognition systems and holds potential applications across diverse domains, such as medical image analysis, object recognition, and autonomous systems.

Acknowledgement

I would like to express my deep sense of gratitude and indebtedness to my high respected and esteemed guide Prof. Mr. Vinay Kumar (Associate Professor, ECE) and Dr. Alok Kumar Singh (Professor, ECE) for having suggested the topic of my project and for giving me complete freedom and flexibility to work on this topic. They have been very encouraging and motivating and the intensity of encouragement has always increased with time. Without their constant support and guidance, I would not have been able to attempt this project. I extend my sincere thanks to all my friends who have been patiently helped me directly or indirectly in accomplishing this project successfully.

AKANSHA LAL
2K21/VLS/03

Contents

ABSTRACT iii
1 INTRODUCTION 1
1.1 MOTIVATION 1
2 BACKGROUND AND RELATED WORK 4
2.1 LITERATURE REVIEW 4
2.2 OVERVIEW 6
2.2.1 Deep Learning 6
2.2.2 Artificial Neural Networks 7
2.2.3 CNN 8
2.2.4 Dilated Convolution 17
2.2.5 Backpropagation 18
2.2.6 Computer Vision 19
3 CONVOLUTIONAL NEURAL NETWORKS 21
3.1 IMAGES STORED IN COMPUTER 21
3.2 ARCHITECTURE 23
3.3 HIERARCHICAL TRANSFER LEARNING 25
4 MODEL TRAINING 27
4.1 HARDWARE / SOFTWARE USED 27
4.1.1 GPU 27
4.1.2 TPU 27
4.2 KERAS 28
4.3 VGG16 29
5 PROPOSED SYSTEM 30
6 EXPERIMENTAL APPROACH 35
6.1 TRAIN, VALIDATION AND TEST 35
6.2 OVERFITTING AND UNDERFITTING ..... **Error! Bookmark not defined.**
6.3 BATCH SIZE 35
6.4 ALGORITHM 36
7 EXPERIMENTAL RESULTS 39
7.1 HEATMAP 39

7.2 ACCURACY COMPARISION 39
7.3 LOSS GRAPH 40
8 CONCLUSION 43
REFERENCES 44

List of Figures

1. Machine learning, Deep learning and Artificial Intelligence
2. Neuron highlighting the chain structure between the axon and dendrite
3. The artificial neural network architecture (ANN i-h 1-h 2-h n-o)
4. CNN 5 layers architechture
5. Applying filter in input image
6. Linear Activation function
7. Sigmoid Activation Function
8. Tanh Activation Function
9. ReLU activation function
10. Softmax Activation Function
11. Applying 2x2 max-pooling on input
12. Applying 2x2 Average pooling on input
13. Fully connected dense layer
14. Neural network with many convolution layer
15. Dropout layer in the neural network
16. Dilated Convolution
17. The gradient descent
18. A 3-D RGB matrix. Each layer of the matrix is a two-dimensional matrix of red, green or blue pixel values
19. The CNN architecture.
20. The filter slides over the input and performs its output on the new layer
21. Types of pooling
22. VGG16 model architecture

23. System architecture

24. 5 layers 32 batch size CNN

25. Heatmap on testing data
26. accuracy vs epoch in CIFAR10

27. accuracy vs epoch in ImageNet

28. Loss vs epoch in CIFAR10

List of Tables

CHAPTER

1 INTRODUCTION

In every electronics devices, Artificial intelligence (AI) has made things interesting and easy. But the philosophy behind that AI is not simple. It takes Computer hours of training to learn those skills that humans don’t give importance.

Machine learning is an application for AI which provides systems to automatically learn, and update based on the previous experience without requiring additional programmed. The computer programs are programmed such that it can access data and learn from it.

They look at the pattern in data and make decisions based on the examples provided. The main aim of the computer is to train it so that it can learn on itself without human intervene and accordingly adjust actions.

ML can also be understood as process of automating and improving based on experiences without any human aid ie programming. They start with good quality data and training computer by machine learning models using different algo.

Machine learning is nothing but a geometrical problem which includes regression and classification.

Various tasks are enabled by AI, including computer vision, which entails the processing and analysis of images with the aim of emulating human vision. Image classification is among the primary objectives of computer vision, where labels or categories are assigned to images. For example, in the case of images containing multiple objects, the task involves categorizing them into classes such as "car," "plane," "ship," or "house."

1.1 MOTIVATION

Image classification is a type of image processing technique that uses image feature information to identify various objects. As science and innovation truly want personal satisfaction has developed, picture robotized order innovation has been applied to an assortment of improvement regions. The customary picture arrangement technique can't precisely grasp the interior connection between the acknowledgment objects while characterizing the picture. Furthermore, the conventional strategy has constraints in the element articulation of the acknowledgment object because of the information's unnecessary trademark aspect, which results in under ideal trial results. This study presents a technique for picture identification in view of convolutional brain networks considering the first. The exploratory calculation utilizes convolutional brain organizations and profound learning. The deep convolution neural network model can be utilized for both element learning and picture arrangement, as opposed to past techniques for ordering pictures.

Disadvantages of conventional CNN:

- There must be a lot of training data, and the object's location
- The difficulty in bringing the problem to the network's attention.
- Unable to respond spatially to incoming data. There needs to be a lot of training data.

Image classification and object recognition are two common applications of convolutional neural networks. Utilizing either filter or a feature extractor, the convolution layer is in charge of extracting essential features. A deep neural network was used to categorize pictures for the CIFAR-10 dataset. A lot of hyper parameters need to be changed in order to get the most out of the network's performance. To limit over fitting, we endeavored to diminish model intricacy by dropout while expanding test sum through information increase. Because it has a direct impact on the network's convergence speed, selecting the right function optimizer is crucial. The network has utilized the Adam optimizer. The organization utilizes the ReLU actuation capability wherever besides in the result layer, which utilizes the Softmax enactment capability. One hot encoding is expected to change over the result names, which are whole numbers, into classifications. Each forward pass in training results in an output value that is compared to the actual output and used to calculate a loss function. The slope of the misfortune capability is spread once again into the

organization to improve the loads utilizing the cross-entropy misfortune capability and Softmax initiation at the result. This system is rehashed for a foreordained number of ages to achieve the most elevated level of accuracy.

CHAPTER

2 BACKGROUND AND RELATED WORK

2.1 LITERATURE REVIEW

A. KNN-based Ensemble of Classifiers:

To accomplish predominant outcomes, we develop an outfit of classifiers and inspect the exhibition of a few classifiers on the CIFAR-10 dataset. We exhibit in CIFAR-10 that, when joined, K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN) are totally unrelated for specific classes. We join a CNN with Principal Component Analysis (PCA) to lessen KNN over fitting and upgrade precision. Our methodology makes our best CNN model better.

B. Image Classification in the Era of Big Data: A Distributed Deep Representation Learning Approach

The ever-evolving demands of the big data era have surpassed the capabilities of traditional image classification techniques. With the availability of a vast number of labeled datasets and the advancement and promotion of high-performance computers, deep learning has transitioned from theory to practice. This research focuses on the image classification method based on distributed deep learning, specifically selecting distributed deep learning tools based on the Flash community platform as the subject of investigation.

To address the challenges posed by the labeled structural deep network embedding (LSDNE) model, which involves a large number of hyperparameters and high model complexity, this study proposes a semi-supervised network that considers neighbor structure learning. The inspiration for this model comes from the Locally Linear Embedding (LLE) algorithm. By incorporating information about the neighbors of each node during network representation learning, the model constructs the next layer of representation based on the physical node and its neighboring nodes. A novel approach called Structural Informed Locally Distributed Deep Nonlinear Embedding (SILDDNE) is proposed by incorporating node properties into the existing Structural Labeled Locally

Deep Nonlinear Embedding (SLLDNE) model. The method for effectively combining the primary features of the node with its properties is comprehensively explored. The learned labels are then sorted using an SVM classifier, and the deep neural network is constructed by integrating network topology, labels, and node properties using SILDDNE.

Experimental results on the CIFAR-10 and CIFAR-100 datasets for standard image classification tasks demonstrate that the proposed network exhibits high generalizability and robust classification performance.

C. Validation of the STL-10 dataset through the utilization of a proficient CNN model trained on CIFAR-10:

An area of AI known as ANN (artificial neural networks) manages calculations that are roused by the design and capability of the cerebrum. A convolutional neural network (CNN) is a kind of significant cerebrum network that is most often used to look at creative pictures. It utilizes a multi-facet perceptron differentiation that is made to require little preprocessing. The CIFAR-10 dataset comprises of 60000 variety 32x32 pictures that are sorted into ten classes of 6000 pictures each. It is frequently utilized for CNN validation, testing, and training. In this post, we developed a model with numerous hidden layers and convolutions that is well-organized. The accuracy of the training is 90.91%, while the accuracy of the validation is 83.69%. It was also checked on the STL 10 data set, which was based on the CIFAR 10 data set, and a lot of accuracy was achieved with the testing. CNN deep model researchers will benefit from this paper.

D. Image Classification using DNN with an Improved Optimizer:

SGD and ADAM, two of the most state of the art advancement procedures, are intensely integrated into profound learning enhancement techniques. Since the expected boundary esteem should be modified for every application, stochastic slope based strategies are non-versatile. By and by, as far as speculation, stochastic enhancers outflank versatile techniques; in any case, ADAM and its subordinates can't achieve this without a fast focalized rate in profound brain organizations. We should decrease the wavering of the loads, which is the main driver of the precision drop, to further develop this speculation execution. In this vein, we tried to introduce Mean-ADAM, a variation of ADAM that has expanded the refreshed loads by an outside weight to decrease swaying and accomplish a higher precision rate than any remaining versatile slope methods up to the reproducing's decision. On picture characterization datasets like MNIST, CIFAR10, CIFAR100,

ImageNet, and others, we can fundamentally further develop its speculation execution, empowering it to contend with SGD. With CIFAR10, we had the option to accomplish 82% at 150 ages and 99.49% with MNIST; nonetheless, the ADAM announced 76% and 99.43%, separately.

2.2 OVERVIEW

This section describes the planned approach for image classification on a deep convolutional neural network model. Designed from scratch specifically for hierarchical architecture CNN.

2.2.1 Deep Learning

Deep learning is subset of ML that contains artificial neural network, that are algorithms inspiredby biological and function of the human brain.

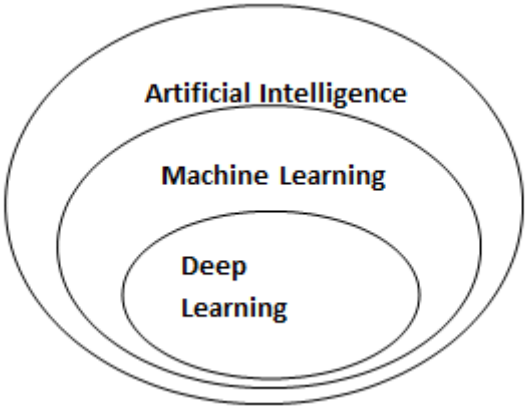


Fig.1: Machine learning, Deep learning and Artificial Intelligence

Deep Neural Networks evaluate data in/ an structured manner to reach comparable results as humans. Deep learning achieves this by putting a algorithms to work with multiple layers known as neural networks.

We may use neural networks to accomplish a wide range of activities, like as, classification, regression and clustering. Based on the patterns between the samples, we may utilize neural networks to categorize / classify unlabeled data. In the classification step, we can train the CNN model on a labeled dataset for categorize the data in this dataset among various categories.

2.2.2 Artificial Neural Networks

Human neural network

By knowing how human neural network works. We will be able to understand about ANN working because it is inspired by human neural network (Figure 2).

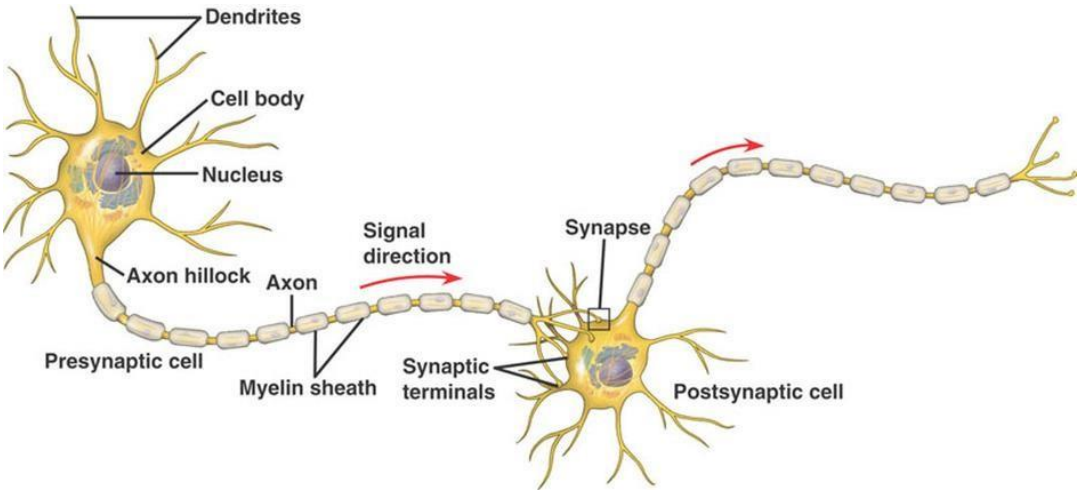


Fig.2: A neuron highlighting the interconnected structure between the axon and dendrite. *(2.)*

The human neural network works in such a way that neuron fires when it is triggered by stimulus (3), the signals thereafter is sent through axon of neuron to its adjacent neuron's dendrites through a connector called synapse. A new neuron fires and causes other neuron leading to fire. This process is repeated in the whole system.

Artificial neural networks

An ANN (artificial neural network) is layers of neuron (can also be called as nodes or units). Each node in a layer has been connected to the adjacent layer's node in a fully connected artificial neural network.

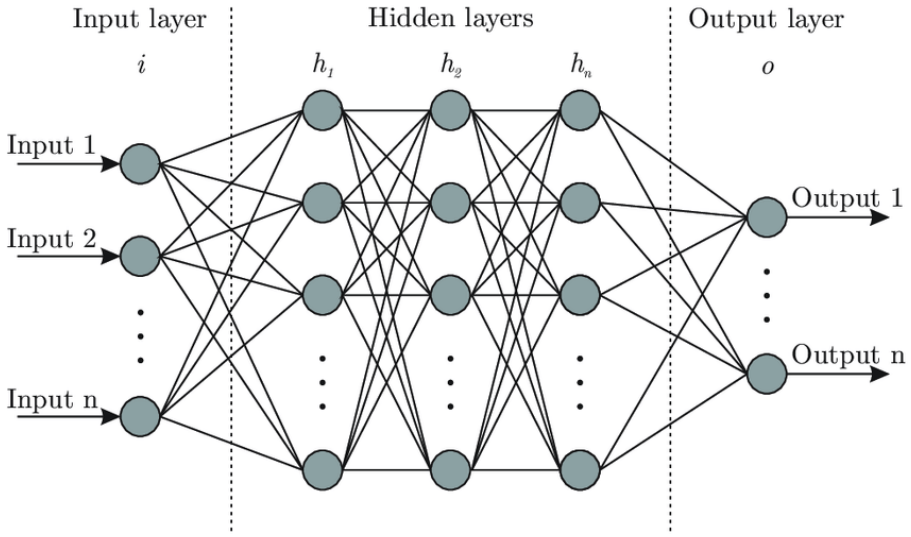


Fig.3. The artificial neural network architecture (ANN i-h 1-h 2-h n-o). (5.)

Input Layer – Feeds all information required, such as images over here.
Hidden Layers – These are the layers between input layer and output layer
A set of different features are detected in each hidden layer. Such as edges and lines, shapes, certain image part (6).
Output Layer – here the predictions are made by the network
Then it is compared by the labels provided.
Back Propagation – where the prediction is incorrect, this technique is used to correct the learning and guess more accurately in next repetition. The labels are made my human. The backpropagation technique is discussed in chapter 2.5 so that it can correct its learning and make correct guesses in next time (7.)

A network makes classifications on their own, without human help after learning enough. (6.)

Artificial neuron

An artificial neuron is a connection point (unit or node) in an artificial neural network and has the capacity to process input signals and produce an outputsignal. (8.)

2.2.3 CNN

Convolution neural network is a one type deep learning model. This neural

network is feed forward neural network where signal flow from output of one neuron to input of next layer neuron means there is no feedback.

A CNN is a technique that takes an image is processed and assigns weights to the features in the image so that they can be discriminated. Compared to other classification techniques, CNNs don'trequire that much preprocessing. The filters can be learned by the convolutions themselves. The architecture of CNN is the structure of neurons as a source of inspiration in the human brain.

In an image, it can capture many temporal and spatial connections. CNNs are capable of performing complex tasks using multi-modal data such as images, text, audio, and video. LeNet, AlexNet, VGGNet, GoogLeNet, ResNet, and ZFNet are examples of ConvNet designs.

CNN uses multiplication of an image matrix with a for extracting features and pre-determined characteristics from it. We use a channel to filter the image and get only the predominant importantfeatures. The images are matrices of pixel values, and the filters are commonly 3x3 or 5x5. The filter is moved across the image with a specified stride, and the values are multiplied and added toprovide a matrix output that is easier to understand.

$$\mathrm{F}(x) = f(f_{n-1}(\ldots f_{\mathrm{i}}(x))$$

Where ‘n’ is number of hidden layer and ‘fi’ is function used in corresponding layer.

In a CNN model there are basically 5 layers.

- Convolution layer
- Activation function layer
- Pooling layer
- Fully connected dense layer
- Predication layer

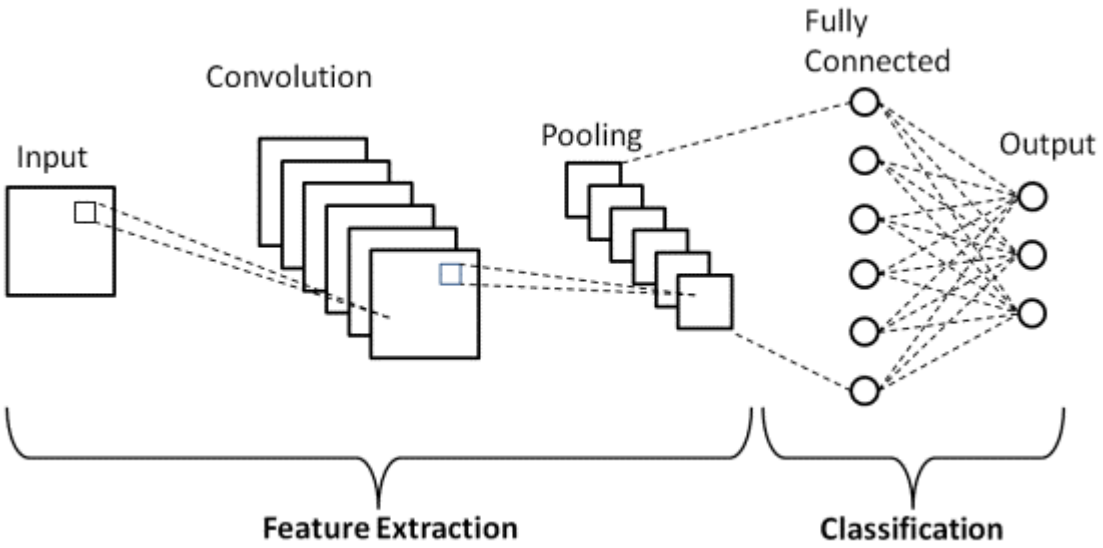


Fig.4: CNN 5 layers architechture

1) Convolution layer

In convolution layer it uses filters size n*n and apply convolution operation all over the imageusing these filters and extract the tiny features of the image which we called feature map. Forexample: In nodule detection of an image VGG16 use to detect edges, shapes, abnormal cells etc.

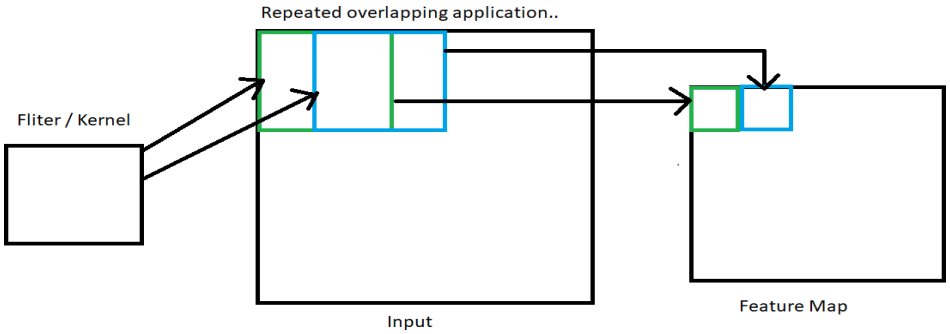


Fig.5: Applying filter in input image

2) Activation function layer

In this layer we use any non linear activation function to bring non-linearity in our model which speeds up training and faster to compute. It is used to learn and understand complexpattern in our data as well asto prevent the numbers from aggregating to zero. Mostly weuse RELU (rectified linear units) as an activation function which can be expressed as

$$f(x) = \max(O, x)$$

There can be other activation function too like sigmoid, tan h and exponential linear units (ELU) etc.

- **Linear activation function**

The equation for a linear function is y= mx, which is the same as the equation for a straight line.

Equation : f(x)=xRange: -∞ to ∞

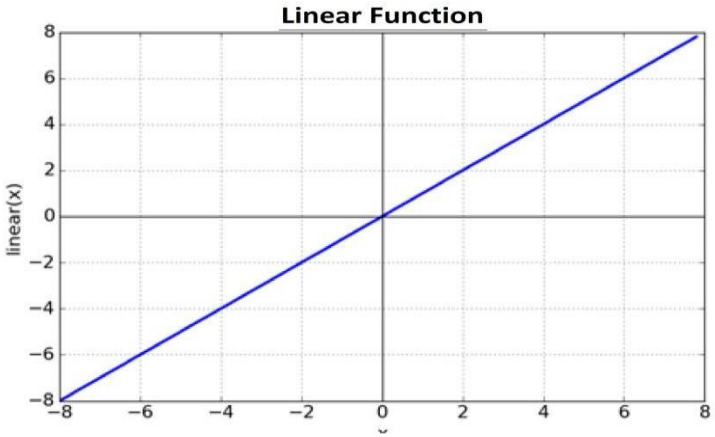


Fig.6: Linear Activation function

- **Sigmoid Function:**

The sigmoid or sigmoid activation function's curve resembles an 'S' shaped curve. Between zero and one is the range of the logistic activation function. Because value of the sigmoid function is limited between zero and one, the outcome is likely to be one if the value is greater than 0.5 & zero else.

Equation: $f(x)=\frac{1}{(1+e^{-x})}$

Range: 0 to 1

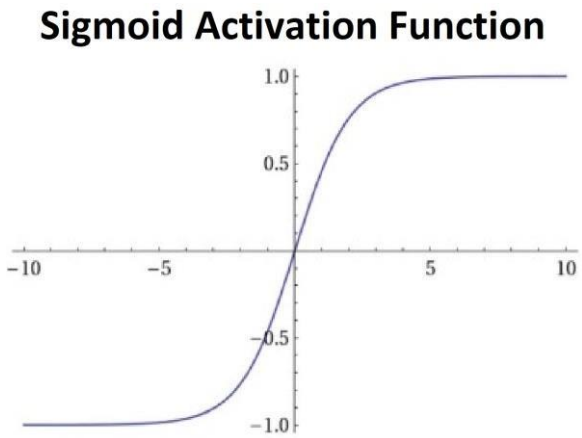


Fig.7: Sigmoid Activation Function

- **Tanh activation function**

Tanh is a hyperbolic tangent function, similar to the logistic sigmoid. The curves of the Tanh and sigmoid activation functions are quite similar, as illustrated in figure 2.13, however Tanh is preferable since the whole function is zero centric.

Equation: $f(x) = \tanh(x) = \frac{-1}{(1+e^{-2x})}$

Range: -1 to 1

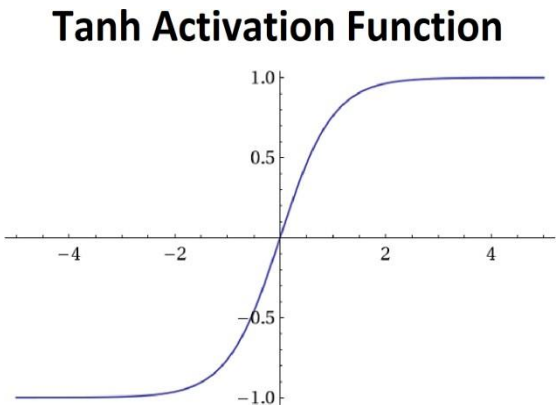


Fig.8: Tanh Activation Function

- **ReLU activation function**

It's most often used activation technique in hidden layers of a deep neural networks. It's themost used activation method in CNN hidden layers. Because the ReLU function is non-linear, we may quickly back transmit errors and trigger multiple layers of neurons. It is less costly than hyperbolic tangent and sigmoid because it uses fewer complex computations. Because just a few perceptron are engaged at any given moment, the cnn is sparse and quick to process.

Equation: $f(x) = \max(0, x)$

Range: [0 to ∞)

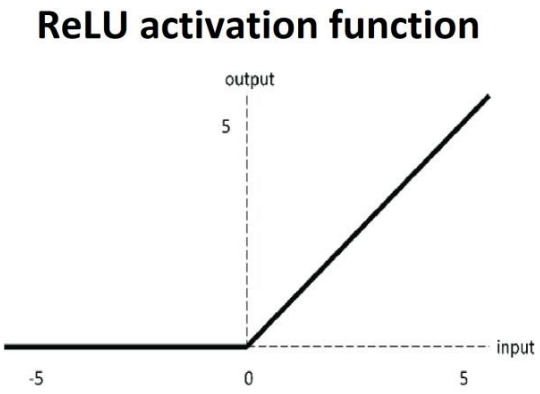


Fig.9: ReLU

- **Softmax function**

It is a function that deals with classification tasks at the fully connected layer. It is commonly employed when dealing with many classes. The softmax function has a range from zero to one. It is best used at the output layer of the deep neural network, where probability is used to characterize the classification from each input.

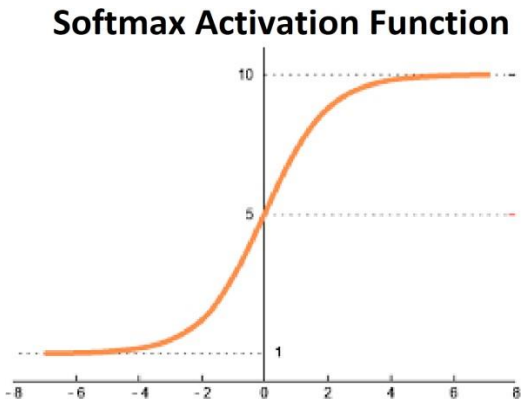


Fig.10: Softmax Activation Function

3) Pooling layer

Pooling is actually used to reduce the size or reduce the dimension of image (feature representation). By experiments it was found that Max pooling is used mostly. In Max pooling we use a window of size m*m and take maximum pixel value among all the pixel values in window of feature map and slides to stride of 'k' and by doing this it cover the whole feature map. There are several types of pooling, such as maximum pooling, minimum pooling, averagepooling.

- Max Pooling:

A method of determining the highest weight of sectors by pooling them together of a feature map and uses it to construct a down sampled (pooled) map of features is called as Max-Pooling. After a convolutional layer, it's typically used. By employing these layers,the dimension of the convolutional matrix is reduced. So, the number of learnable parameters is reduced, as is the network's processing complexity.

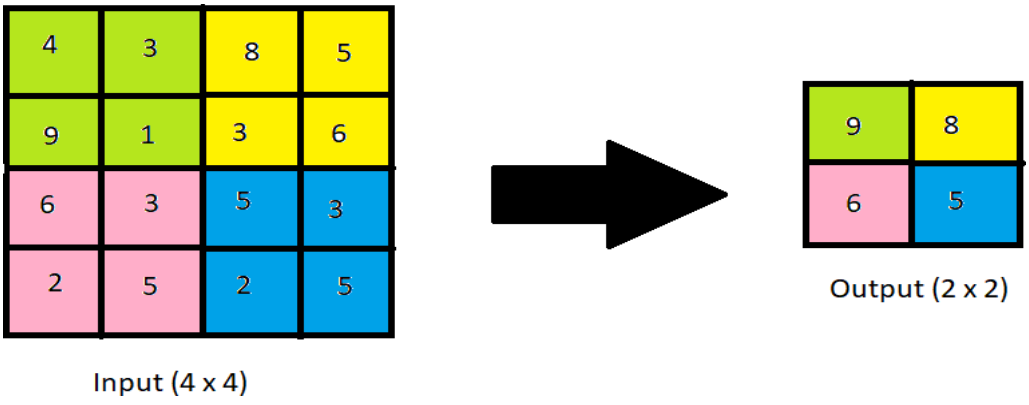


Fig.11: Applying 2x2 max-pooling on input

- **Average Pooling:**

In this, the average of the values accessible in the region of the feature space covered by the kernelis used in it. The pooling layer creates a connection between the convolutional and fully -connected layers in general.

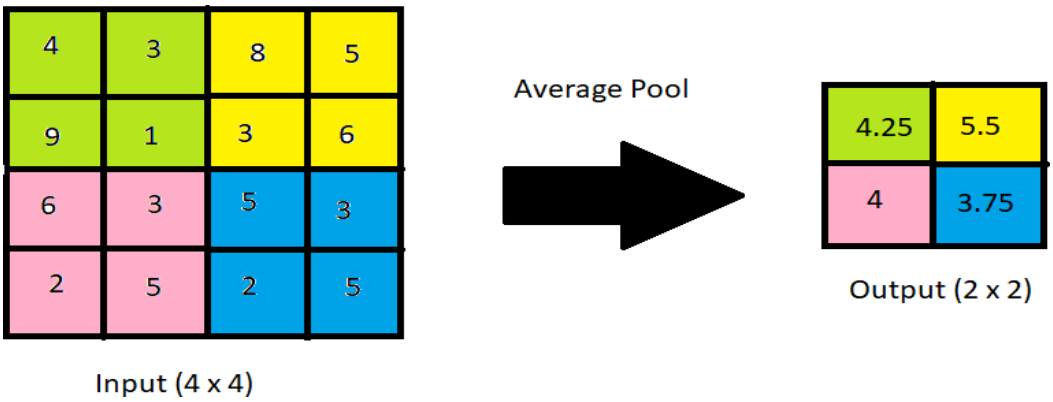


Fig.12: Applying 2x2 Average pooling on input

4) Fully connected dense layer

It is the last layer, where the categorization takes place. We can combine our filteredand shrunk pictures into a single list which is called vector.

In Fully connected dense layer, each neuron is connected to every other neurons of next layer.

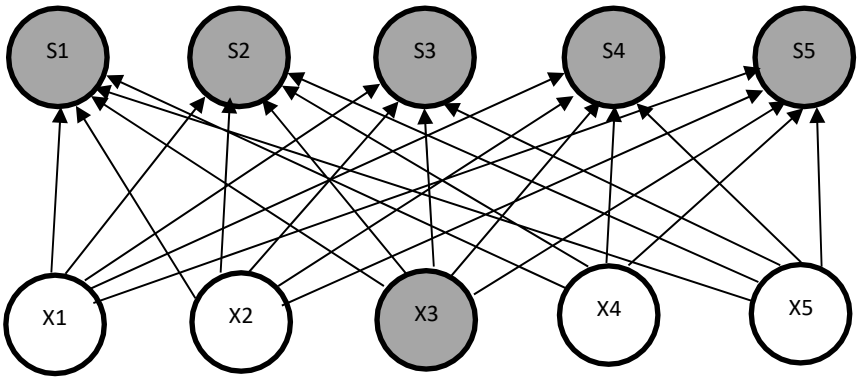


Fig.13: Fully connected dense layer

5) Predication layer - The predication layer is added to last layer of fully

connected dense layer to computeprobabilities of the different classes. Example: Softmax, SVM.

The below given figure shows convolution neural network architecture.

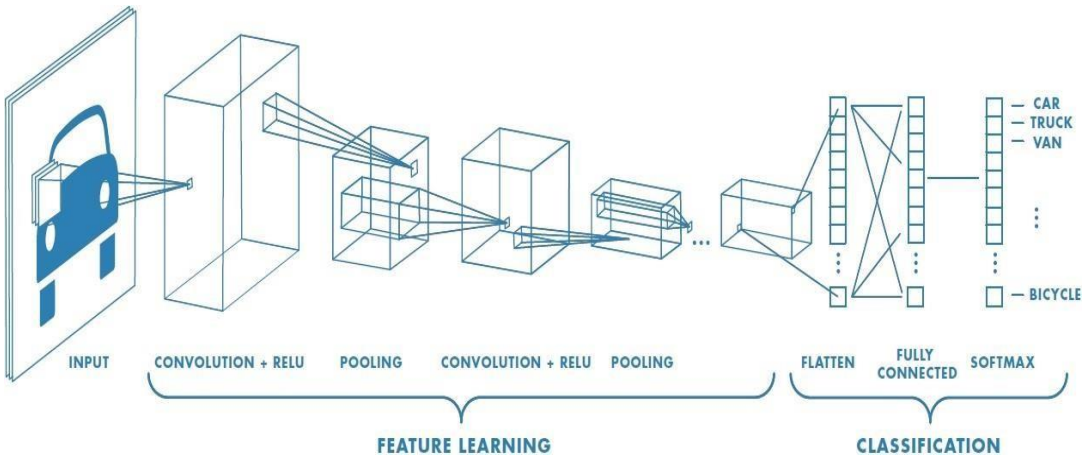


Fig.14: Neural network with many convolution layer [9].

6) Dropout Layer:

When a perceptrons/neurons in a neural network is turned off with a probability P during training, the term "dropout" is used. Assume a probability of P = 0.25, which means that throughout training, 25% of the neurons will be dropped. As a result, a quarter of the neurons with in neural network would not be examined, and the neural network will become easier. To avoid overfitting, we add a dropout layer in the CNN. Simple terms, throughout the training process, a certain number of neurons are discarded from the deep neural network in the dropout layer.

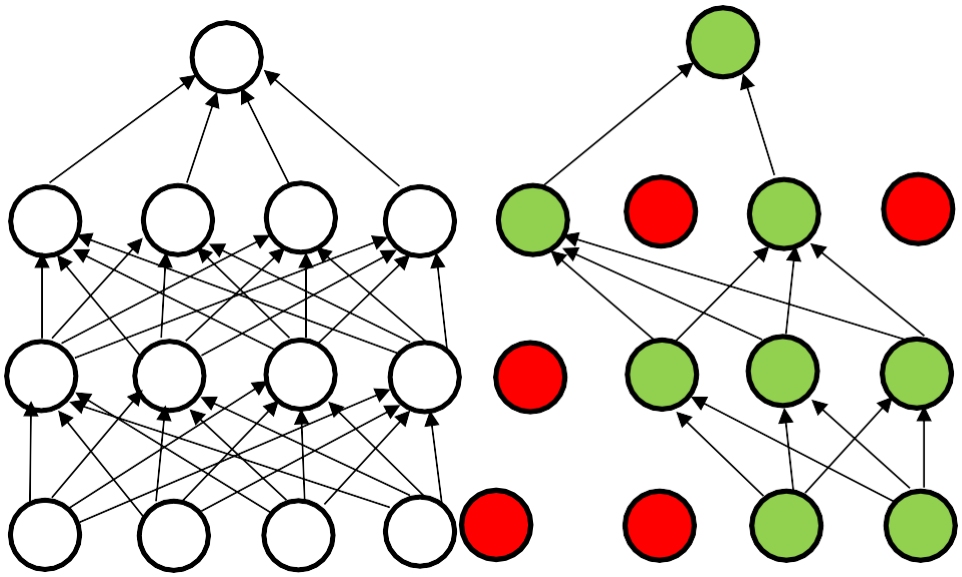


Fig.15: Dropout layer in the neural network

2.2.4 Dilated Convolution

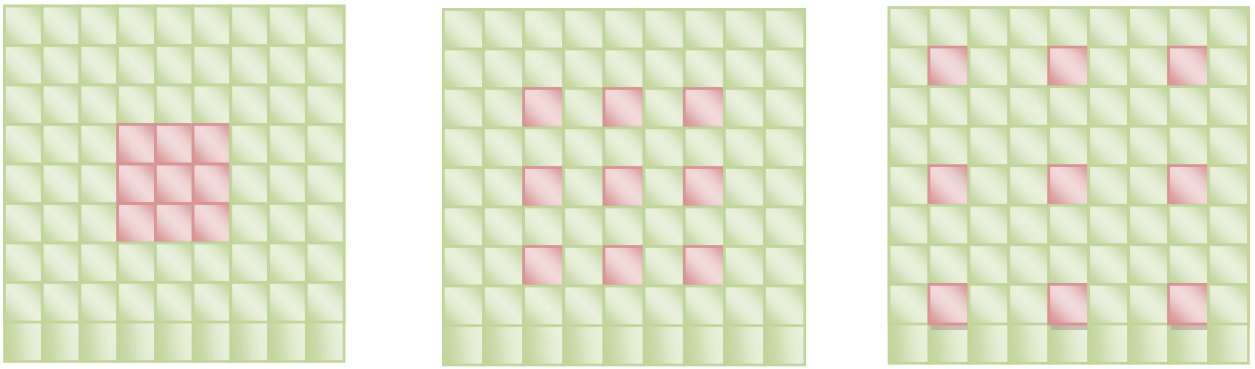
It's a mechanism for expanding a kernel or filter by making holes in between the elements. In layman's words, that's the same as convolution, but now with pixel skipping to cover a broader area of the input. The 'dilation factor' argument specifies just how far the input is stretched. In other terms, the filter skips (dilation factor-1) pixels dependent on the value of dilation rate.

We can get additional information without increasing the number of filter parameters by employing this technique. Dilated convolution allows you to cover a larger region of the input images without pooling. The goal is to extract more details from the output after each convolution layer. At the same computing cost, this approach provides a larger field of vision. We calculate the value of a 'dilation factor' by evaluating how much knowledge is collected with each convolution at different 'dilation factor' values.

Dilated Convolution's Benefits:

- A more expansive receptive field is available.
- Effective in terms of computation.

- Memory usage is lower.
- There is no degradation in the produced image's resolution.
- The convolution's structure assists in keeping the information in order.



Kernel size: 3*3
Dilation rate = 1

Kernel size: 3*3
Dilation rate = 2

Kernel size: 3*3
Dilation rate = 3

Fig.16: Dilated Convolution

2.2.5 Backpropagation

The learning process of neural networks is facilitated by the backpropagation algorithm, which involves predicting and learning from their errors. An explanation of backpropagation using gradient descent is provided in this chapter.

Gradient descent

The gradient descent algorithm is employed to improve the loss function value in a neural network. It operates by iteratively modifying the internal weights of the network in a manner that reduces the loss function value. The weights are assessed by calculating the gradient of the loss function in relation to the weights, and subsequently adjusted by taking a step in the opposite direction of the gradient, scaled by a learning rate. Convergence happens when the decrease in the loss function value becomes insignificant, indicating that the weights have reached a stage where additional adjustments have minimal effect on reducing the loss.

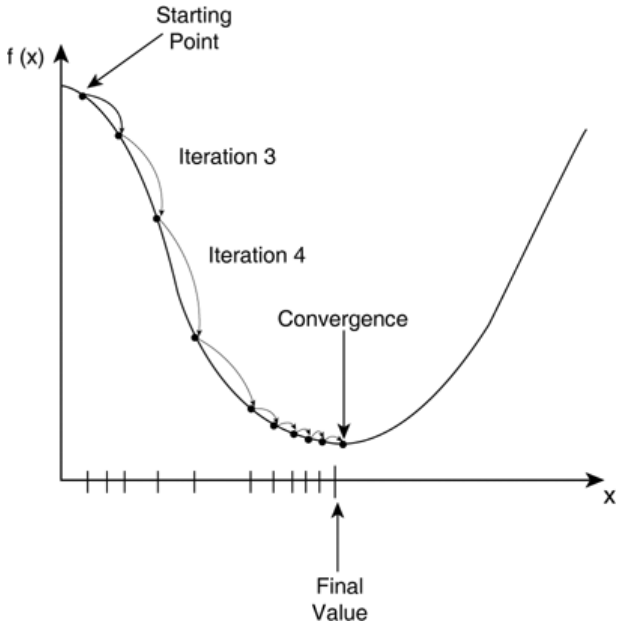


Fig.17: The gradient descent (16.)

Loss function

The loss function is a key component in training a machine learning model, especially one involving neural networks. It calculates the difference between the actual desired output and the neural network's anticipated output. A smaller loss means that the model's predictions are more likely to correspond to the expected outputs, hence the purpose of training is to minimize this loss function. (7).

Learning rate and introduction to the Adam optimizer

The learning rate plays a crucial role in optimization algorithms by influencing the magnitude of the steps taken during each iteration. One such algorithm, Adam, stands out for its ability to calculate personalized learning rates for different parameters of the neural network. This adaptiveness empowers Adam to navigate intricate and high-dimensional spaces with greater efficiency, potentially resulting in quicker convergence and enhanced optimization performance.

2.2.6 Computer Vision

Computer faces problem due to:

- Lens limitation
- Viewpoint variations
- Changing light conditions
- Issues of Scale
- Non-rigid Deformation
- Partially hidden
- Clutter
- Object class variation
- Optical Illusions

Applications of Computer Vision

- Handwriting Recognition
- Image Recognition over 1000 classes
- License Plate Readers
- Facial Recognition
- Self-driving Cars
- Hawkeye and CV Referrals in sports
- Robotic Navigation
- SnapChat Filters

Convolutional neural networks (CNNs), a method based on deep learning, a subset of machine learning, which in turn is a subset of AI, are frequently used for image classification.

The dataset used for this research is CINIC-10, which was acquired from the University of Edinburgh. CIFAR-10 and ImageNet are two well-known datasets for image classification, and CINIC-10 combines their benefits while resolving their drawbacks. Given that it is neither excessively huge nor too tiny, CINIC-10 is intended to serve as a benchmarking dataset (1).

Ten categories have already been assigned to the photographs in CINIC-10: truck, ship, frog, horse, bird, cat, deer, aeroplane, car, bird, cat, and deer. The dataset is split into three equal

subsets, each with 90,000 images: train, validation, and test (1). The goal of this project is to create a convolutional neural network that can classify these images automatically..

Linear classification theory

- Load the data (X and Y)
- Instantiate the model
- Train (“fit”) the model
- Evaluate the model
- Calculate the accuracy on train and test model

How to build a model

1. Architecture of the model (equation to go from input to prediction).
2. How is the model trained?
   Cost/ loss/ error function
   Gradient descent to minimize the cost

CHAPTER

3 CONVOLUTIONAL NEURAL NETWORKS

3.1 IMAGES STORED IN COMPUTER

All images are made up of three colors called pixels, and according to RGB model, each and every pixel contains the three components red, green, and blue colors. Each color element's value range normally ranges from 0 (no color) to 255 (maximum saturated output).

In the field of digital image processing, a coloured image can be represented as a three-layered pixel matrix. Red, Green, and Blue colour channels are each given their own layer. Within the layer, a two-dimensional matrix represents the pixel values for each colour channel.

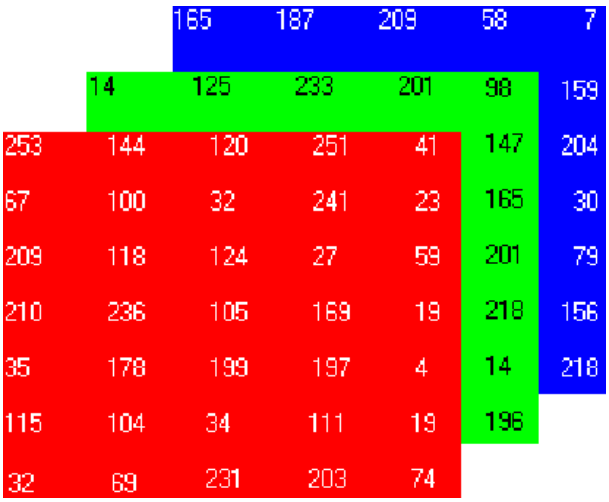


Fig.18: 3D RGB matrix.

Convolutional Neural Networks

An advanced form of neural network architecture called a convolutional neural network (CNN) is created specifically to process and comprehend visual data. Due to their distinctive qualities and design concepts, such as convolutional layers, pooling or subsampling layers, numerous layers, and the capacity to capture hierarchical representations, they excel in image identification tasks. CNNs are frequently used in deep learning architectures for image recognition applications, which has significantly advanced artificial intelligence and computer vision.

3.2 ARCHITECTURE

Among all CNN models, a common architecture is shared (Figure 8) (21).

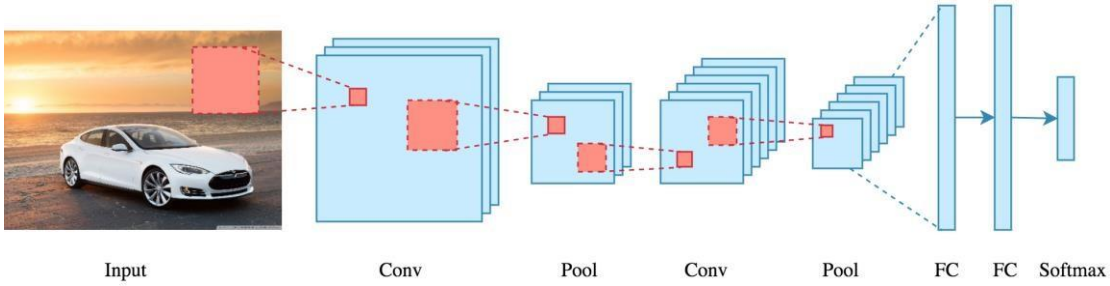


Fig.19: The CNN architecture. (21.)

The VGG16 model used in this thesis receives an input image and applied to a series of convolutional layers (hidden layers) and pooling layers on it. Finally the classifiers, fully connected layers are used to obtain the output after that. In the next chapters, comprehensive explanations of each element of the architecture will be given.

Convolutional layers

Simply said, "convolution" describes the mathematical process of fusing two functions to produce a third function. Two sets of information are combined as a result of this combination.

A convolutional layer, also known as a filter or kernel, is used in Convolutional Neural Networks (CNNs) to analyse the input data, ultimately producing a feature map.

The dot product, or multiplication, between a 3x3 filter matrix and a 3x3 region of the input image's matrix occurs during the convolution process. The output value, also known as the "destination pixel" on the feature map, is then obtained by adding the resulting matrix. The feature map is completed by computing the dot product with each remaining 3x3 segment while the filter iteratively moves over the input matrix.

Multiple filters are applied to a single input to create the feature maps that make up a convolutional layer's final output, which are then combined.

The two techniques in convolutional layers are padding and strides.

The filter's strides define how many pixels it traverses over the input matrix. If there is underfit filter of the input matrix, padding is used. There are two kinds of padding: zero or

"same" padding, which adds zeros to the borders to ensure the filter matches the input matrix, and valid padding, which removes the boundary pixels.

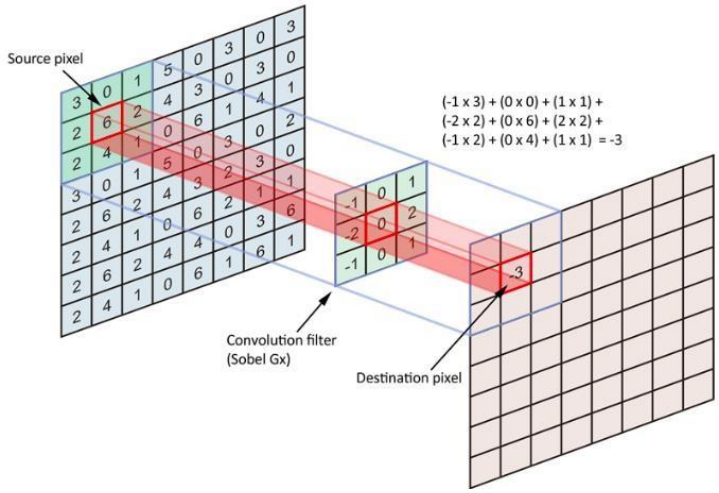


Fig.20: The filter slides over the input and performs its output on the new layer. (22.)

Pooling layers

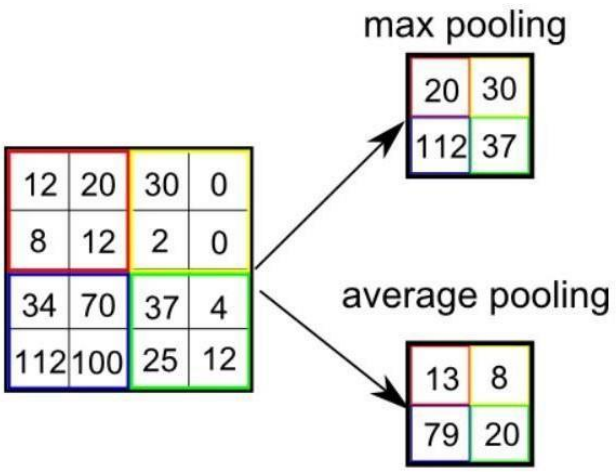
The output of the feature extraction dimensions were decreased by pooling layers. The width and the height is kept the same keeping the depth same. This mapping is useful because it allowed for the dominating feature extraction from the feature maps by using less processing capacity for data processing.

The feature maps' dimensions were decreased by pooling layers, notably their height and breadth, while maintaining their depth. This decrease was advantageous since it allowed for the extraction of the dominating features from the feature maps while using less processing capacity to process the data.

Max pooling and average pooling were the two different kinds of pooling layers.

in Figure 21.

Average pooling gives the average value of the elements, while max pooling outputs the



maximum value in the area of the image covered by the filter. Max pooling was deemed more efficient since it was more effective at extracting prominent characteristics. (25.)

Fig.21: Types of pooling. (25.)

Fully connected layers

It is the last layer, where the categorization takes place. We can combine our filtered and shrunk pictures into a single list which is called vector. In Fully connected dense layer, each neuron is connected to every other neurons of nextlayer.

It is used as classifiers in the model where the final classification done. The input vector in the form of flattened column vector is fed to classifier is fully connected layers, which are the same as the fully connected ANN architecture

A Dense layer, which is other name for fully connected layers is given through an activation function (e.g., tanh/ReLU), but the output Dense layer is allowed through Softmax. In this type of multiclass classification, the loss function used is Cross Entropy.

The output obtained from the Softmax function is an N-D vector,

where N is the number of classes of the CNN.

3.3 HIERARCHICAL TRANSFER LEARNING

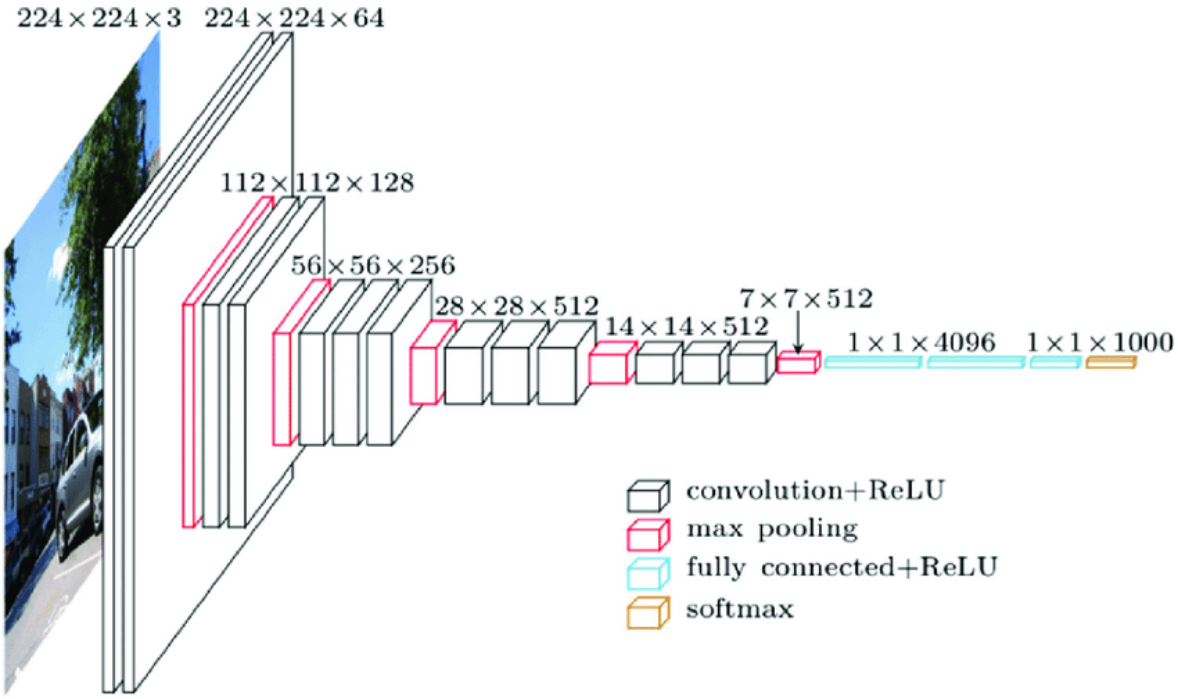


Fig.22: VGG16 model architecture

CHAPTER

4 MODEL TRAINING

4.1 HARDWARE / SOFTWARE USED

The thesis uses free GPUs from Google Colab (Colaboratory). The deep learning framework used is Pytorch framework.

4.1.1 GPU

The graphics processing unit, or GPU, can be used for both personal computing and business-related computing. It has made most important place in computing technology. It has been designed to use for parallel processing.
GPU was initially designed to render 3D graphics but with time became flexible and the capabilities are improved.
The developers also started harnessing the computational power of GPUs to expedite workloads in deep learning and high-performance computing. Some of the applications are

- Gaming
- Video editing and content creation
- Machine Learning

4.1.2 TPU

The Tensor Processing Units or TPU are ASICs (Application Specific Integrated circuit) developed by Google to accelerate workloads of machine learning. It trains model for performing large matrix operations using hardware found in machine learning algorithm. TPU has high-bandwidth memory on-chip for larger models and large batch sizes. Its matrix processing (MXP) is specialized for neural network specific workloads. It has thousands of multiply-accumulate operators that are

directly connected in form of large physical matrix. A high-computational throughput can be achieved using TPU. XLA (accelerator linear algebra) compiler used by TPU host machine.
A single Cloud TPU has four chips with one or two tensor cores in each chip

Best hardware to be used for available workload distribution can be known by keeping following points in mind

CPU

Maximum flexibility with quick prototyping
Small models with small batch size
Tensorflow operations written in C++
Limited I/O available

GPU

Models that have significant amount of custom TensorFlow operations
Larger batch sizes

TPU

Models with dominated matrix computations
Models that required weeks or months to get trained
Large Batch sizes

4.2 KERAS

Keras is a deep learning framework or API (Application Programming Interface) used to implement neural network by Google. It provides frontend high level abstraction using python with multiple backend options for computational purposes.

The backend framework supported by keras are

- Tensorflow
- Theano
- CNTK

4.3 VGG16

1. The torch package's CIFAR10 dataset comprises 60,000 images with 10 labels, each having a size of 32x32 pixels. By default, when using torchvision.datasets.CIFAR10, the dataset is divided into 50,000 images for training and 10,000 images for testing.

2. VGG16, an extensively researched and developed convolutional neural network by Karen Simonyan and Andrew Zisserman, is a deep network that may pique your interest.

3. Transfer learning is a process that involves reusing a pre-trained model to meet the requirements of developers or data scientists. In this particular case, I utilized the VGG16 model to address the CIFAR10 dataset.

CHAPTER

5 PROPOSED SYSTEM

In this project, Google Colab was the primary working environment that was utilized. The first step involves specifying the machine used for training the model, which can be either cuda or cpu. Afterward, the selection of the number of epochs, batch size, and learning rate for the training process is performed. As mentioned previously, the CIFAR10 dataset contains 10 labels, and these labels are stored in the variables associated with the class.

To improve CNN generalization performance during the early learning phase for image categorization, we present a brand-new hierarchical Transfer Learning. This is crucial for real-time applications that require excellent Transfer Learning generalization performance in a short amount of time. The proposed model cans scalable combine data from numerous data sources for Transfer Learning. On the Image Net and CIFAR-10 datasets, we tried out our method. The findings demonstrate that the proposed hierarchical Transfer Learning significantly accelerates training, with an average and equal increase in testing accuracy for the CIFAR-10 scenario and an improvement in accuracy for the Image Net case in the early stages of learning.

Advantages

- Without the need for human intervention, it automatically identifies significant characteristics. When it comes to picture identification challenges, it has a very high accuracy. It's easy to understand and quick to use.
- Capacity to make a two-layered inside portrayal.

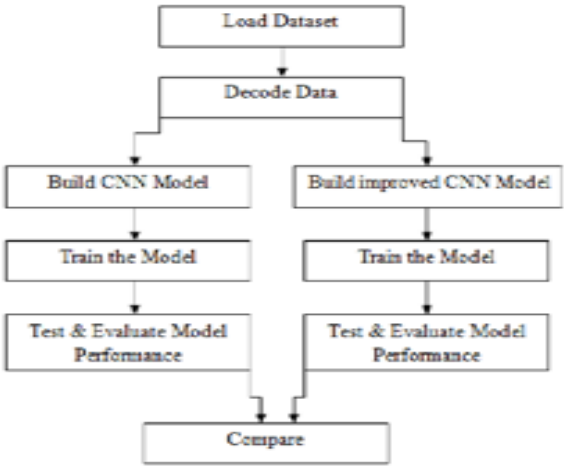


Fig.23: System architecture

Deep learning, a form of ML, enables network to acquire knowledge through practical experience, simulating human learning processes. This technology has proven essential in enabling self-driving cars to identify pedestrians, lampposts, and stop signs. Furthermore, voice control features found in various consumer devices like smartphones, tablets, televisions, and hands-free speakers rely on deep learning. Unsurprisingly, deep learning has gained considerable attention in recent times, achieving results that were considered unattainable. In deep learning, the models learn to do classification tasks immediately from images, text, or audio. In some cases, the accuracy of deep learning models surpasses that of humans. Training these models involves utilizing a significant amount of labeled data and complex neural network architectures.

MODULES:

1. Picture information
2. picture pre-handling
3. picture division
4. extraction of highlights
5. data preparation and testing
6. Deep learning algorithm

Collecting data: Data collection heavily relies on datasets, which are curated collections of digital images used by developers to test, train, and evaluate the effectiveness of their algorithms. Obtaining data can be achieved through various methods such as online scraping, acquiring data from third-party sources, purchasing datasets from resellers, and more. Autoencoders show optimal performance when applied to visual data. The system offers support for file type filters and includes compatibility with Bing.com filterui filters. The downloading process utilizes multithreading and allows for adjustable thread pool sizes. It is also possible to collect only the URLs associated with the images.

Data Cleaning: Data cleaning, commonly referred to as the process of rectifying or removing inaccurate, corrupted, improperly formatted, duplicated, or incomplete information within a dataset, represents the primary method employed. When combining data from multiple sources, various forms of duplication or mislabeling can occur. The practice of identifying and correcting inaccurate, fragmented, duplicate, or otherwise erroneous data within a data collection is known as data cleaning or data scrubbing. It entails identifying errors in the data and resolving them through modifications, updates, or deletions.

Feature Extraction:

An underlying assortment of crude information is partitioned and decreased into additional reasonable classes through the course of dimensionality decrease, which incorporates include extraction. Subsequently, the cycle will be made more straightforward. The enormous number of factors in these tremendous informational indexes is the main component. To handle these factors, a lot of computational power is required. By choosing and combining factors into highlights, include extraction assists with choosing the best component from huge informational indexes, eventually lessening information volume. While precisely and extraordinarily depicting the genuine informational collection, these qualities are easy to process. Picture Handling - One of the best and most interesting spaces is picture handling. You will essentially begin interacting with your photographs in this section to better understand them.

Model training:

Simplify and plan. We must first consider how the computer perceives the images.

Collect. Obtain the most diverse and variable training dataset possible for each task. Upload and sort. You are prepared with your images, and now it is time to sort them.

- Accurate and trained.
- Utilize torch vision to load and normalize the CIFAR10 test and training datasets.
- Characterize a Convolutional Brain Organization.
- Characterize a misfortune capability.
- Utilize the preparation information to show the organization.
- Utilize the test information to test the organization.

Testingmodel:

In this module, we examined the output of the trained deep learning model by evaluating it with the test dataset. This evaluation involves a specific type of test that generates clear images.

Performance Evaluation:

Within this module, we employ various enhancing performance evaluation criteria, such as accuracy, and classification error, to gauge the performance of the trained deep learning model. As the score increases, the model's detection accuracy improves. Model evaluation, a common practice, involves utilizing different assessment metrics to comprehend the strengths, weaknesses, and overall performance of an ML model.

Detection:

Object detection involves identifying and locating all potential occurrences of actual-world objects, such as cars, flowers, and faces, within images or videos, with real-time precision. This technique employs learning algorithms and derived features to detect and classify instances of specific object categories. Firstly, an input image is taken and divided into multiple regions. Each of these regions is then treated as an individual image. These images are then passed through a CNN (Convolutional Neural Network) and categorized into different classes.

CHAPTER

6 EXPERIMENTAL APPROACH

6.1 TRAIN & VALIDATION

A dataset is used to train the model, helping the neural network to learn its weights and biases. Following the prediction phase, the model undergoes evaluation using the validation dataset to fine-tune its hyperparameters. Ultimately, the fully trained model is assessed for its performance by evaluating it using the test dataset.

6.2 INACCURATE MODEL PREDICTION

When the noise is present in data and is detected by model, overfitting occurs. Essentially, the model exactly fits the data, resulting in excessive dependence on the training data. Whereas, underfitting arises when the data trend cannot be captured by the model. Both overfitting and underfitting lead to inaccurate predictions when applied to new datasets.

6.3 BATCH SIZE

In most of the cases, it can be impossible to feed the whole dataset at the same time in a neural network, necessitating its division into parts or batches.
The number of training samples in a single batch is indicated by the batch size (30).

Epoch: The neural network is fed with the entire dataset (i.e. every training sample) once from forward and backward. (30).

Dropout: The application of Dropout can help mitigate overfitting. Dropout involves randomly omitting nodes and connections during the training process (31, abstract).

Batch normalization: Batch normalization can also be employed for the purpose of reducing overfitting. It involves modifying the activations and scaling them, thereby

optimizing the input layer. The mathematical details of batch normalization will not be discussed in this thesis. (32.)

```
classes = ('plane', 'car' , 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck')

batch_size = 32
```

Fig. : classes are defined

6.4 ALGORITHM

CNN: The convolutional layering of learned characteristics with input data in a CNN architecture makes it ideal for processing photos and other two-dimensional data. CNNs eliminate the need to manually extract features, removing the need to identify image-classification characteristics. The CNN directly extracts characteristics from photos. There is no pre-training for the essential characteristics; All things being equal, the organization trains on a bunch of pictures to instruct them. In light of programmed highlight extraction, profound learning models are very exact for PC vision applications like article order.

Utilizing tens or many more secret layers, CNNs figure out how to perceive various visual highlights. The visual qualities that have been learned become more muddled with each secret layer. For example, the primary secret layer could figure out how to perceive edges, while the last layer could figure out how to perceive more convoluted structures that are one of a kind to the thing's math.

Stage 1 for CNN's Following stages:

Stage 2: Pick a Dataset

Stage 3: Set up the Dataset for Preparing Get information on preparing.

Stage 4: Sort the Dataset.

Stage 5: Designate Names and Components.

Stage 6: includes normalizing X and changing over names into absolute information.

Stage 7: Separate the X and Y tomahawks so CNN can utilize them.

Neural networks are composed of artificial neurons, forming complex structures capable of processing multiple inputs and generating a single output. This capability is a fundamental aspect of brain organization, which involves transforming input into valuable output. In many cases, convolutional neural networks (CNNs) comprise numerous convolution and pooling layers that aid in deep learning, allowing the

network to identify significant features in visual information such as photographs. Generally, a neural network has many layers such as an input layer, an output layer, and one or more than one hidden layers. Due to this diverse arrangement, CNNs excel in recognizing spatial, rotational, and translational invariant features. The basic components of a CNN model are illustrated in the provided diagram. A single image is represented as a pixel-value tensor, and convolution layers assist in extracting the image's features (forming feature maps).

6.5 CODE SNIPPETS

```
[ ]  # check GPU availability
     device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
     print(device)

     cuda:0
```

Fig. : The used hardware is GPU to run the model

```
# Downloading and Preparing the Dataset
"""
You may observe that one of the transforms is resizing the images to 224x224 size. Well, this is because t
"""
transform = transforms.Compose([transforms.Resize((224, 224)), transforms.ToTensor(), transforms.Normalize

trainset = torchvision.datasets.CIFAR10(root='./data', train=True, download=True, transform=transform)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=32, shuffle=True)

testset = torchvision.datasets.CIFAR10(root='./data', train=False, download=True, transform=transform)
testloader = torch.utils.data.DataLoader(testset, batch_size=32, shuffle=False)
```

Fig. : The size of image is resized to 244 * 244 and the batch size is 32

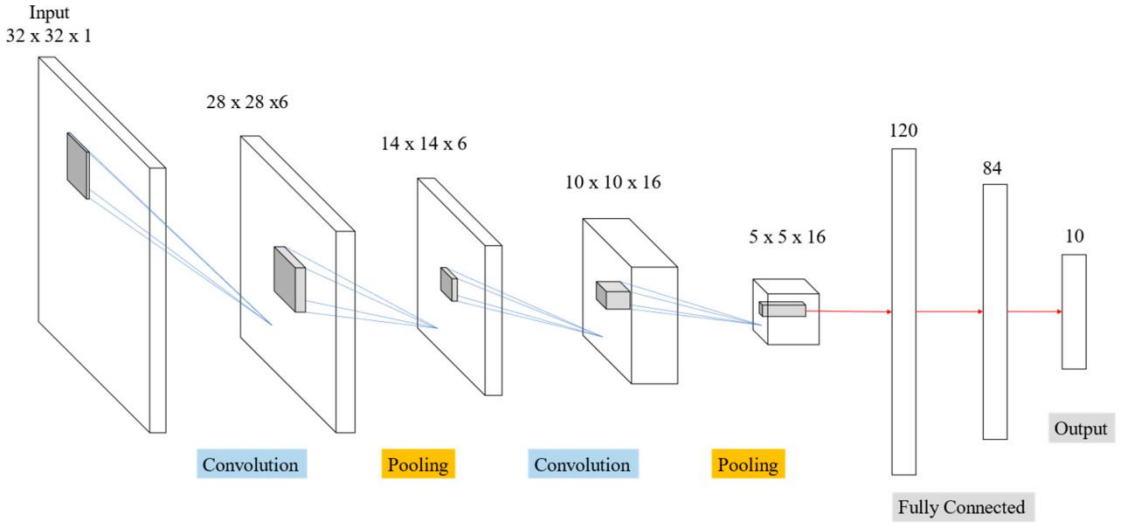


Fig. : 5 layers 32 batch size CNN

```
# Downloading the VGG16 Network
vgg16 = models.vgg16(pretrained=True)
```

Fig. : The model used is pre-trained

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
```

Fig. : Feature extraction in VGG

```
(avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
```

Fig. : Pooling in VGG model

```
(classifier): Sequential(
  (0): Linear(in_features=25088, out_features=4096, bias=True)
  (1): ReLU(inplace=True)
  (2): Dropout(p=0.5, inplace=False)
  (3): Linear(in_features=4096, out_features=4096, bias=True)
  (4): ReLU(inplace=True)
  (5): Dropout(p=0.5, inplace=False)
  (6): Linear(in_features=4096, out_features=1000, bias=True)
)
```

Fig. : Classifier which is modified

CHAPTER

7 EXPERIMENTAL RESULTS

7.1 HEATMAP

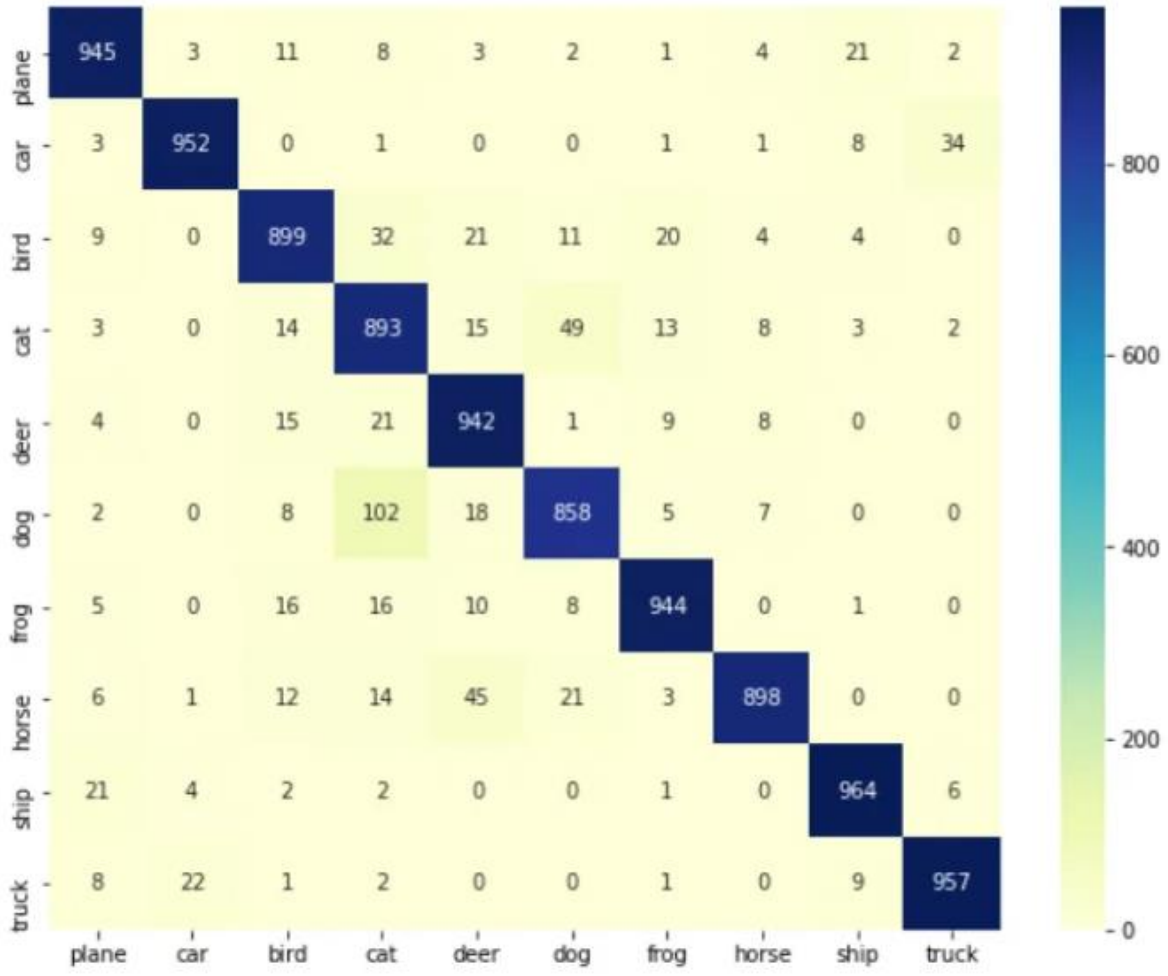


Fig.25: Heatmap on testing data

7.2 ACCURACY COMPARISION

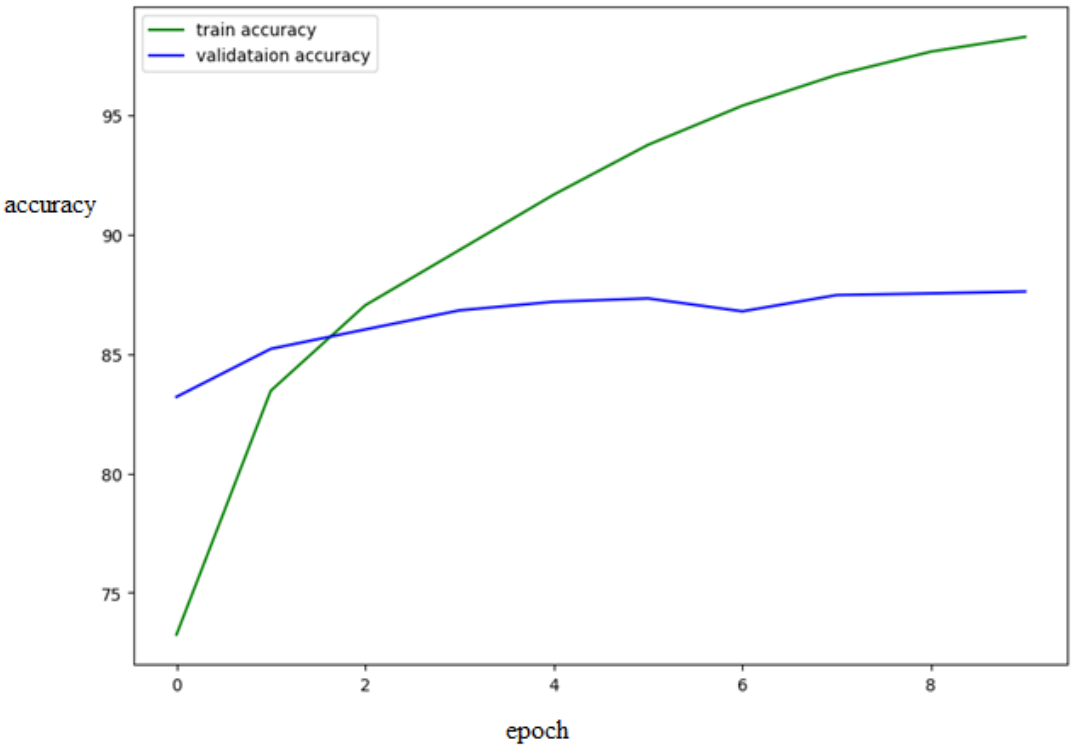


Fig.26: accuracy vs epoch in CIFAR10

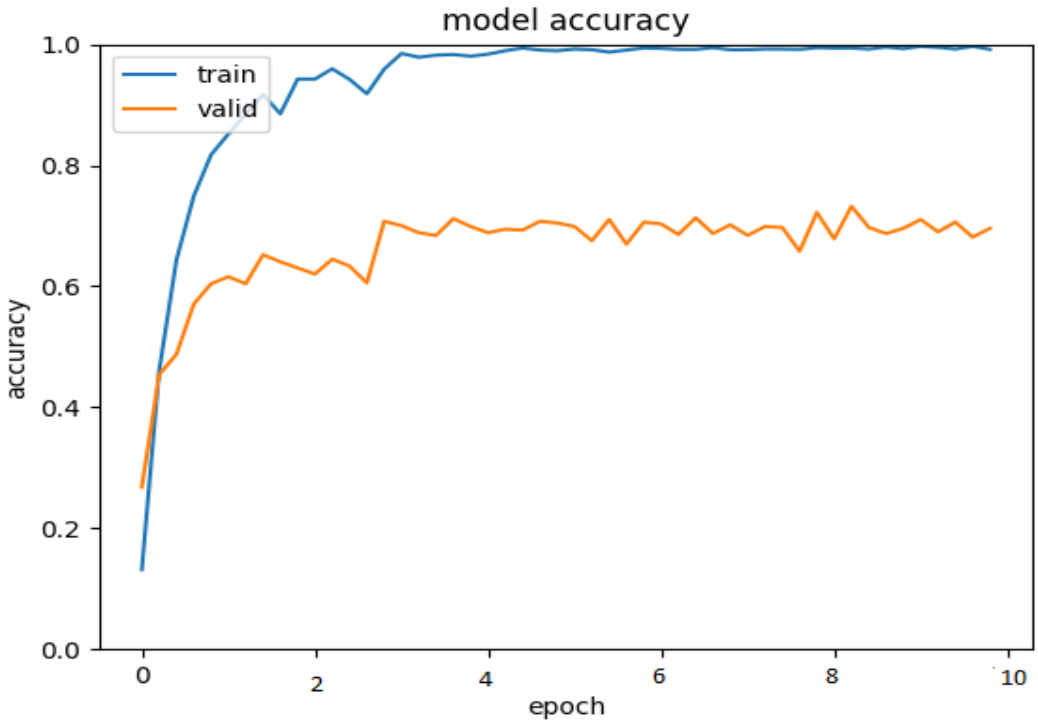


Fig.27: accuracy vs epoch in ImageNet

7.3 LOSS GRAPH

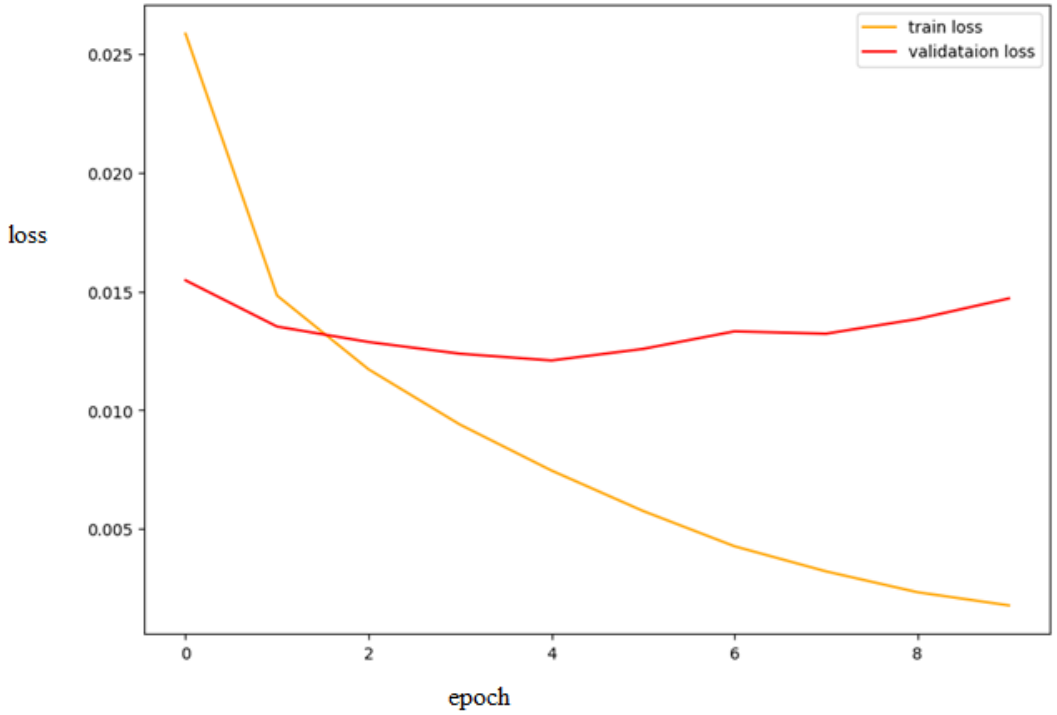


Fig.28: Loss vs epoch in CIFAR10

```
epochs = 10
```

100%| 1563/1563 [05:22<00:00, 4.84it/s]
Train Loss: 0.0259, Train Acc: 73.25
100%| 313/313 [01:02<00:00, 4.99it/s]
Validation Loss: 0.0155, Validation Acc: 83.20
100%| 1563/1563 [05:15<00:00, 4.96it/s]
Train Loss: 0.0148, Train Acc: 83.46
100%| 313/313 [01:03<00:00, 4.97it/s]
Validation Loss: 0.0135, Validation Acc: 85.21
100%| 1563/1563 [05:14<00:00, 4.96it/s]
Train Loss: 0.0117, Train Acc: 87.03
100%| 313/313 [01:02<00:00, 5.01it/s]
Validation Loss: 0.0129, Validation Acc: 86.02
100%| 1563/1563 [05:14<00:00, 4.98it/s]
Train Loss: 0.0094, Train Acc: 89.35
100%| 313/313 [01:02<00:00, 5.01it/s]
Validation Loss: 0.0124, Validation Acc: 86.82
100%| 1563/1563 [05:14<00:00, 4.97it/s]
Train Loss: 0.0075, Train Acc: 91.66
100%| 313/313 [01:02<00:00, 5.04it/s]
Validation Loss: 0.0121, Validation Acc: 87.18
100%| 1563/1563 [05:22<00:00, 4.85it/s]
Train Loss: 0.0057, Train Acc: 93.75
100%| 313/313 [01:09<00:00, 4.48it/s]
Validation Loss: 0.0126, Validation Acc: 87.32
100%| 1563/1563 [05:20<00:00, 4.88it/s]
Train Loss: 0.0043, Train Acc: 95.39
100%| 313/313 [01:03<00:00, 4.91it/s]
Validation Loss: 0.0133, Validation Acc: 86.78

100%|████████████| 1563/1563 [05:16<00:00, 4.94it/s]
Train Loss: 0.0032, Train Acc: 96.68
100%|████████████| 313/313 [01:03<00:00, 4.97it/s]
Validation Loss: 0.0132, Validation Acc: 87.46
100%|████████████| 1563/1563 [05:15<00:00, 4.96it/s]
Train Loss: 0.0023, Train Acc: 97.65
100%|████████████| 313/313 [01:02<00:00, 4.99it/s]
Validation Loss: 0.0138, Validation Acc: 87.53
100%|████████████| 1563/1563 [05:15<00:00, 4.96it/s]
Train Loss: 0.0018, Train Acc: 98.27
100%|████████████| 313/313 [01:02<00:00, 4.97it/s]
Validation Loss: 0.0147, Validation Acc: 87.61
63.44018760919571 minutes

CHAPTER

8 CONCLUSION

Utilizing the most well-known preparing and test datasets, CIFAR10 and CIFAR100, the review inspected the expectation precision of three unmistakable convolutional neural networks (CNN). In each dataset, we just saw ten classes. Our critical goal was to choose the precision of the various associations on the comparable datasets and to assess the consistency of figure by all of these CNN. For the purpose of assessing the effectiveness of the networks for a variety of items, we carried out a comprehensive prediction study. It is essential to keep in mind that when the scene is detected and recognized by the network, complex frames frequently result in confusion. Also, it was found that, notwithstanding the way that beds, couches, and seats are unmistakable and effectively conspicuous in reality, the precision paces of the prepared organizations shifted because of perplexity. The results showed that transfer learning-trained networks performed better than existing ones in terms of accuracy. Numerous studies and works have been done on it. It is easy to use and can be used on a variety of systems, offering a wide range of applications. The organization will be unable to be prepared on customary work area work because of the equipment necessities, however it tends to be prepared and the fitting model created with not very many prerequisites.

REFERENCES

[1] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 5 2015

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in CVPR09, 2009.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/ 4824-imagenet-classification-with-deep-convolutional-neural-networks. Pdf

[4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[6] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories,” in 2004 Conference on Computer Vision and Pattern Recognition Workshop, June 2004, pp. 178–178

[7] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, “Deep feature extraction and classification of Hyperspectral images based on Convolutional Neural Network (CNN)” IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232-6251, 2016.

[8] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Neural Inf. Process. Syst., Lake Tahoe, NV, USA, 2012, pp. 1106–1114

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015

[10] Hara, K., Jagadeesh, V., & Piramuthu, R., “The role of deep convolutional neural network and pose-dependent priors.” In Proceedings of IEEE Winter Conference on Applications of Computer Vision, 1-9, 2016

[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E., “Imagenet classification with deep convolutional neural networks.” Advances in Neural Information Processing Systems, 1097-1105, 2012

[12] Simonyan, K., & Zisserman, A., “Very deep convolutional networks for large-scale image recognition.” arXiv preprint. arXiv:1409.1556., 2014