Hochschule Fulda
University of Applied Sciences

A thesis presented for the degree of
MSc in Global Software Development

Deep Learning Methods For Vehicle Classification

Farrukh Ahmed Khan
147727

Supervised by
Prof. Dr. Alexander Gepperth
Applied Computer Science

Co-supervised by
Prof. Dr. Yvonne Jung
Media computer science and computer graphics

May 2019

Abstract

Cars are one of the most valuable and important means of transportation bringing us the flexibility in moving from one place to another. In our modern life cars are playing a beneficial part which seems to be never-ending. In addition, cars are also a subject of interest which people take personally nowadays from buying the newest car with the newest technology keeping in mind the color, design, and comfort to the modification of the design with fancy things. The need for the car has shifted to the highest level in the past years. Due to its enormous advantages, the automotive companies are on the race to reach on top and designing and manufacturing the models year by year.

Since the problems aims to highlight vision related tasks revolving around "Vehicles". We use Deep Learning Methods to classify vehicles from number of classes. The famous custom CNN's architecture were used for the experimentation and comparison.

The data-set used for this specific problem was provided by CompCar [45] having more than 20000 car images with 163 car makes (a.k.a manufacturers) and 2004 car models.

Statutory Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other University.

....

Date and Location Signature

Acknowledgement

I would like to deeply thank my thesis supervisor Prof. Dr. Alexander Gepperth, who supports me in every challenging moment. the completion of this thesis could not have been possible without Prof. Dr. Alexander Gepperth. I would also like to thank my thesis co-supervior Prof. Dr. Yvonne Jung for sitting on our panel.

Last but not the least, I would like to express my profound gratitude and thanks to my parents- Mr. and Mrs. Khan, without them none of this would indeed be possible.

Thank you.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

Cars are one of the most valuable and important means of transportation bringing us the flexibility in moving from one place to another. In our modern life cars are playing a beneficial part which seems to be never-ending. In addition, cars are also a subject of interest which people take personally nowadays from buying the newest car with the newest technology keeping in mind the color, design, and comfort to the modification of the design with fancy things. The need for the car has shifted to the highest level in the past years. Due to its enormous advantages, the automotive companies are on the race to reach on top and designing and manufacturing the models year by year.

Due to several rich and unique properties cars gives researchers an opportunity on a range of enormous research topics. Because of its many features, cars can help to foster Computer Vision problems and Image Processing Algorithms. The appearance of a car such as its design, the manufacturer, the model year opens lots of door for the researchers such as classification of cars, attribute prediction, for example, seating capacity, maximum speed, and displacement. In comparison with Human Face Verification which is a very interesting topic amongst the researcher, the car verification system which targets whether two cars belong to the same model or not is getting popular in research areas.

As getting deeper into it, the viewpoints of different car models tend to be more challenging than face verification, and as it can be used in many useful applications for example categorization of cars on paying tolls to automatically categorize the type of vehicle for paying taxes. Similarly, Vehicle classification or verification can be used for verifying the model and manufacturer of the car if the number plate recognition fails [45]. The applications for vehicles are enormous and can be useful in many places, such as traffic counting, analysis for intelligent traffic, think of an application that can give the user a piece of detailed information about the car or any vehicle when you take a picture of it. A low-cost camera can do these amazing things, instead of using expensive laborious hardware.

Despite having such an enormous amount of applications and practical interest, car model analysis only attracts a few attention, and that is because of the lack of a high-quality data-set of vehicles. The more you go deeper into cars, you will notice the more amount of data-set with the high quality you require and the more challenging it gets. This thesis is about the classification of cars on the high-quality data-set provided by "The Chinese University Of HongKong", which is named as

"Comprehensive Cars" with "Comp Cars" being short, containing more than 200000 images of more than 2000 models. The Comp Cars data set contains two types of data set with different scenarios "Web-Nature" and "Surveillance-Nature" [45].

This thesis focuses on the data given by CompCar [45] whether the data is of high caliber and quality for Deep Learning Algorithms as mentioned above.

The classification techniques used in this paper are based on using Deep Learning Algorithms to classify images from 163 car makes and 2004 model names.

In [45] the authors use Convolutional Neural Networks for classification, verification, attribute predictions from car make (a.k.a manufacturers) and model names. The Convolutional Neural Networks are the Deep and Machine Learning algorithms or architectures that are widely used for classification of images. These Networks are designed to require minimal pre-processing.

A tight bounding boxes used to process all the data used, to get the perfect center of the images. Our approach compares the deep learning algorithms used in [45] with the addition of new CNN architecture VGG19 which gives more better results than AlexNet and Inception(Google-Net) used in the above-mentioned paper.

1.1 Motivation

Due to the dynamic nature of vehicles, the need for vehicle identification and classification has became more important and challenging in recent years, from security point of view to the classification of the car in the parking lot or verification of stolen vehicle [4] , likewise counting of cars for traffic planning and analysis, the applications are enormous.

The deep learning methods are one of the powerful and emerging technology nowadays, which gives the developers an opportunity by applying the architectures on the data and get the predictions which can be very useful. A great example or application using Machine Learning or deep learning methods would be predicting the popularity on the basis of the appearance of the car also recommending the similar styles of cars can be beneficial for the firms and the user or customers.

Similarly having a useful and clean data of car models can lead to more challenging problems such as attribute prediction from car models.

1.2 Goals Of Thesis

The main goals of this Master Thesis are listed below:

- As mentioned earlier, the thesis will be based on Deep Learning Methods or Algorithms which uses multiple layers of Convolutional Neural Networks (a.k.a CNN) architecture on the data "Comp cars", the hierarchy of tags which are comprised of car make names as parent label and car model names as child similarly going more into the depth of the data the released year of the car in the hierarchy is used as the child of car model name.

- Before designing and applying custom CNN architecture layers the famous Deep Learning Architectures used in the paper [45] were applied on the given data i.e. ALexNet[14], Inceptionv3(GoogleNet)[34]
- A new deep learning architecture that was applied to the given data i.e. VGG19 which tends to provide more better results than the two mentioned above.
- Lastly, among these networks, precisely one architecture will be selected that showed the potential to address the problem of this thesis and custom CNN architecture will be built on top of the selected network.
- The proposed architecture of CNN are applied in two different ways. i.e. Training of models to predict car makes (a.k.a manufacturers) only, and Training of models to predict car models only.
- The image data for this thesis will be split into smaller sets. For instance, Top-5 classes of the car make i.e. the number of image data set which is higher than 4000. Top-13 classes of the car make i.e. the number of image data set having more than 2000 samples and lastly, Top-23 classes of the car make i.e. the number of image data set having more than 1000 samples.

The important thing which was kept in mind while designing custom CNN architectures and applying deep learning algorithms is that they will only be trained and provide predictions on the data set which depicts exactly one car in the image. Therefore, the Deep Learning trained models might not perform on the data having the number of cars in the image. Although, it can only be done by preprocessing of the images i.e. cropping the images with bounding boxes or detecting the cars using computer vision segmentation techniques and then cropping every image with the bounding boxes. Although, this approach is not a part of this thesis due to limited time constraints.

1.3 Thesis Structure

The structure of this Master Thesis is organized in the following chapters

- **Chapter 2** describes the Foundations of Neural Networks. It also describes Artificial Intelligence, Deep Neural Networks, Convolutional Neural Networks as well as the CNN architectures that are used in this Thesis. Furthermore, it also describes the description of the data set that is being used to support this Thesis.
- **Chapter 3** is about the data set preparation and Image pre-processing methods, the data is split for different scenarios such as the number of classes with more than 4000 samples, number of classes with more than 2000 samples, and number of classes with more than 1000 samples each. In Image pre-processing, the number of steps and methods applied to the given image data to make it readable for deep learning architectures are discussed.
- **Chapter 4** will discuss about the implementation of Deep Learning Network Architectures of CNN (a.k.a Convolutional Neural Networks or ConvNets) in detail, the Network Architecture and the Technology stack that has been used

and also the Training part of each model i.e. Training on AlexNet, InceptionV3 and VGG19, and lastly the testing part will be presented.

- **Chapter 5** is about the experimentation and evaluation part of the thesis, in which ConvNets training will discuss in detail.
- Lastly, **Chapter 6** discuss the conclusion and the future work will be presented.

1.4 Related Work

As discussed earlier object classification and verification is a hot-topic in researchers nowadays. Following this line of research, many studies and research have proposed different types of datasets on various categories, including birds [38], flowers[24], cars [45], dogs[18] etc. Below are some of the related research.

1.4.1 Related Research-I

One of the most closely related research on which this thesis is based on car recognition from images is done by researchers from **The Chinese University Of Hong Kong**. They released a Large-scale high quality cars data set named as "Large-Scale Car Data Set for Fine-Grained Categorization and Verification of cars" [45]. In their research the scientists studied three applications using CompCars data set i.e. fine-grained car classification, attribute prediction, and car verification.

The Authors of the CompCars select the total of 78,126 images and divided them into three subsets.

- **Part-I:** The first subset contains more than 400 car models, 431 to be precise with the total of 30,955 full car images and 20,349 images of the attributes of each car.
- **Part-II:** The second subset (Part-II) contains 111 car models with more than 4,000 images in total.
- **Part-III:** Lastly, the third subset (Part-III) contains 1,145 car models with 22,236 images.

In **"fine-grained classification"** the scientists tried to classify images from 431 car models. The image data set of cars were grouped with respect to viewpoints of the car which are as follows: (F) Front, (R) Rear, (S) Side, (RS) Rear Side, and (FS) Front Side and All-View. Viewpoint is the identification angle of each car image. For the above application, the researchers applied a Convolutional Neural Network, specifically Over-feat Model which is pretrained on ImageNet classification task. With the results it was concluded that **"All-View"** point images gave the best results as compared to all other viewpoints. Below is the detailed table of results or accuracy for each viewpoint.[45]

Viewpoint	F	R	S	FS	RS	All-View
Top-1	0.524	0.431	0.428	0.563	0.598	0.767
Top-5	0.748	0.647	0.602	0.769	0.777	0.917
Make	0.710	0.521	0.507	0.680	0.656	0.829

Table 1.1: Fine-Grained Classication results

Where "Make" denotes the car make (a.k.a manufacturer) classification accuracy.

In **"Attribute Prediction"** the researchers tried to predict the attributes of the given car from a proper viewpoints. The attributes were differentiated by states i.e. 2,3,4,5 doors and seat number were also has four states i.e. 2,4,5,¿5. By applying fine-tune CNN model, it was concluded that attribute "Number Of Doors" has the best accuracy as compared to all other attributes.[45]

In **"Car-Verification"** the Authors perform car verification following the pipeline of Face Verification [19]. The testing data was divided into three sets, each of which have different difficulty level named as, "Easy", "Medium" and "Hard". The "Easy Set" is selected from the same viewpoint, the "Medium Set" is selected from random viewpoints, whereas the "Hard Set" was selected from the negative pairs. The negative pairs (-1) is "Uncertain Viewpoint". It was concluded that the verification accuracy of the same viewpoint is higher than other i.e. "Easy Set".

1.4.2 Related Research-II

Another interesting research on Vehicle Classification that came across is "Vehicle Classification using Transferable Deep Neural Network Features" [48]. In this research the author uses two methodologies.

- **Vehicle Detection From Road Image**: The images of the vehicle were taken from a static camera from a Motor way. All of the image data, that was taken was comprised of the rear side of the vehicles.

Figure 1.1: Road Image Dataset Example

The steps taken for extraction of the vehicle are as follows

- Median Filter
- Background subtraction
- Removal Of Noise
- Otsu's Method for detecting background and foreground.

- **Vehicle Classification using AlexNet features**: The images obtained from Vehicle Detection were used as the input images for classification. The classes of the vehicles were classified into two classes: i.e. passenger class and other class. Where the passenger class includes SUV, Sedan, and MPV and other class includes vehicles like van, truck or any other types of vehicle. The difference between both classes were not distinctive. Examples of the classes are shown below in the figure.

Figure 1.2: Example of the two classes

Where (a) and (b) in Figure 1.3 were considered as **"Passenger"** classes where as (c) and (d) were considered as **"Other"** classes. Deep Convolutional Neural Networks were used for the development and AlexNet [14] were used for the feature extraction. The vehicle then passed to AlexNet model after re-sizing of images obtained from above section. After extraction of the features from fc6(Fully Connected Layer6) and fc7(Fully Connected Layer7) of the model, Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) were used for dimensional reduction. It was concluded that LDA gives better result than PCA and also SIFT-FV-based methods.

Accuracy	LDA	PCA
Alexnet-fc6	97.00	96.45
Alexnet-fc7	96.80	96.10
SIFT-FV	92.30	91.30

Table 1.2: Accuracy Comparison

1.4.3 Related Research-III

The third research was carried out by Dongin Zhao, Yaran Chen amd Le Lv and named as **"Deep reinforcement learning with visual attention for vehicle classification"** [47]. They propose a novel CNN model which named as CNNVA abbreviation of **Convolutional Neural Network Model based on Visual Attention** to recognize the vehicle. The model uses reinforcement learning to determine the visual attention which can calculate the key areas of the vehicle in the image. The summary of there research is as follows.

- Using reinforcement learning algorithms such as DQN (Deep Q Network), SARSA etc they propose a method which can be use to find key areas to help vehicle classification
- To determine the next view point of the vehicle. they combine both information entropy and reinforcement learning.

The detailed architecture of the proposed model i.e. CNNVA is shown in the figure 1.5 below.

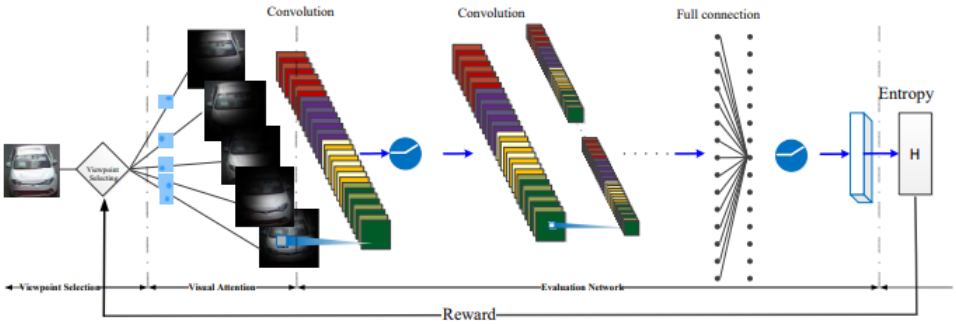


Figure 1.3: The Architecture of proposed Model CNNVA

Chapter 2

Foundations

2.1 Artificial Intelligence

Artificial Intelligence (AI) is an intelligent machine that has power to think, to analyze and to make decisions. In simple words, AI is a programmable human brain inside a machine, that has power to make decisions based on the conditions that have given to it, to reach the required goal. It is the combination of processes by machines, which includes learning, reasoning and self-correction. Artificial Intelligence is one of the most interesting topic in researchers nowadays. The power of Artificial Intelligence is immerse. Some of the particular applications of AI are expert systems, speech recognition, and machine vision. Artificial Intelligence can be categorized by two types [27]

- Weak Artificial Intelligence: Weak AI also known as **"Narrow Artificial Intelligence"**, is a system that is only designed for some particular goals. such as Apple's Siri is a form of Weak AI.
- Strong Artificial Intelligence: Strong AI is a system which is able to find solutions on its own without any kind of human intervention.

2.1.1 History Of AI

Artificial Intelligence is founded by John Mccarthy [20], together with Allen Newell, Marvin Minsky and Herbert A. Simon. John Mccarthy is also known as the **"father of AI"**, who coined the term Artificial Intelligence in 1955, and in 1956 AI emerged as a field of Information Technology. Inspired by Allan Turing in 1950, who introduces a famous "Turing Machine" by putting his ideas into action and by testing whether **"Machines Can Think?"**, which gained a lot of fame and after series of testing which is known as **"Turing Test"** or **"Imitation Game"** as it was called in the paper, it was concluded that it is possible for the machines to think and learn like humans.

Since it's discovery, the AI industries has gone through many hype cycles. In previous decades, failing to deliver the promised goals, or over-promising leads to the failure of AI era, which is commonly known as **AI Winter**.

AI Winter is said to be the period when most of the interest and funding in the AI field was vanished. As said earlier, due to unreachable goals, and promises that didn't go successful, AI fell into the bottom.

As AI has gone through many hype cycles, and there may have been a lot of AI Winters occurred in the past decade, but two of the famous or we should say main periods of **AI Winters**, which are known as First and Second AI Winters are discussed below [5].

- **The First AI Winter:** The first AI was started in early 1970's. At the time, the scientist have made all sorts of promises, that the AI era will be in every corner in the world, that failed drastically. The thinking states that AI will soon conquer humans whether in playing games like chess or automate translations. None of these went on success and as a result the interest and funding in the field of AI effected a lot, because of which the first AI Winter started.
- **The Second AI Winter:** The second AI winter was started in 1980's by one of the famous computer system known as **"Expert Systems"**. Expert systems were designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as if-then rules. Expert systems were likely to move on the right direction, having success in many sectors of AI. But instead of solving general problems, they were focusing their power on domain specific problems. Expert systems got a lot of attraction and interest as well as funding from tech companies and public sector.The interest was international with the Fifth Generation Computer Systems Project in Japan. But similarly, Expert Systems got failed on delivering on the promises. And Japan's economy crumpled badly, in the result Japan could not came near to influence the tech industry for years.

2.1.2 Rise Of Artificial Intelligence

As the hype of AI created in the early 1950's to 1980's, most of the AI's breakthrough aren't noticeable to most of the people. That can be seen in the following graph of attendance at National conference of Artificial Intelligence. It is clear that the downfall of U.S. based AI research was started happening after 1980's. [32]

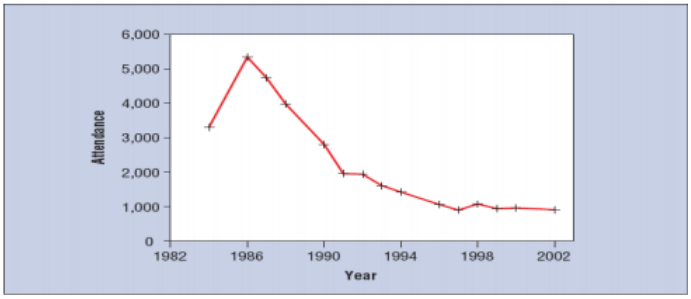


Figure 2.1: Attendance at the National Conference of AI

As the applications of AI world expanded during years, the rise of Artificial Intelligence gets triggered again, with advancement of Machine Learning and Deep Learning in the late 2000 which considered as subset of Artificial Intelligence. A small explanation about AI and it's subsets are given below.

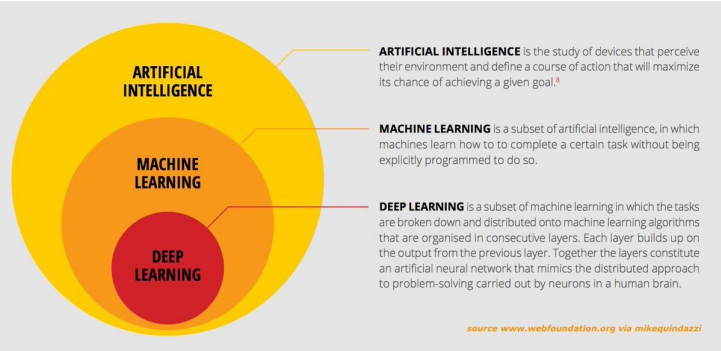


Figure 2.2: Artificial Intelligence and Subsets [27]

With the advancement of Artificial Intelligence to its subsets i.e. Machine and Deep Learning, the researchers and academic institutions begin to take more interest which can be seen in the following research graph taken by AIINDEX.ORG. [7]

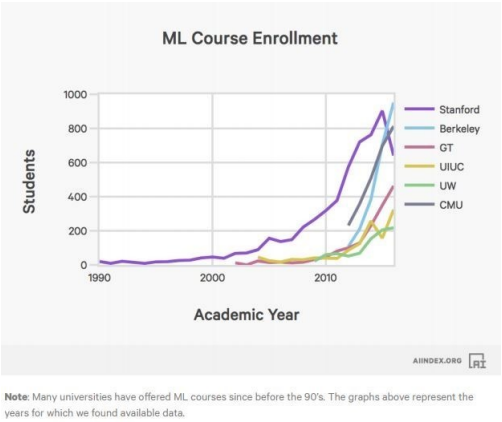


Figure 2.3: Rise Of AI in Academic Institutions

Similarly, As the hype of AI begins again the research on AI has increased 9x since 1996. The following graph illustrates the publishing of AI research. [7].

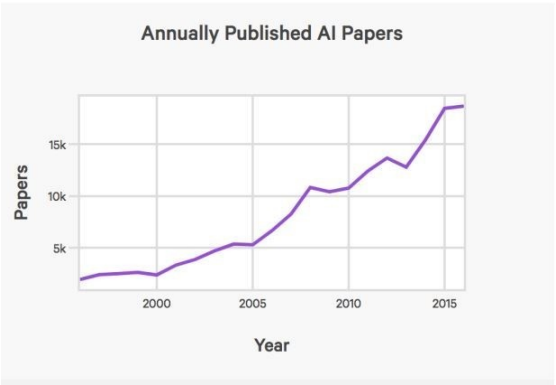


Figure 2.4: Annually Published AI Papers

Moreover, The funding raised by Artificial Intelligence increased exponentially during recent years. A technology and research powered firm known as "Venture Scanner" tracked the data for total funding raised by AI start-ups for each year. [28]

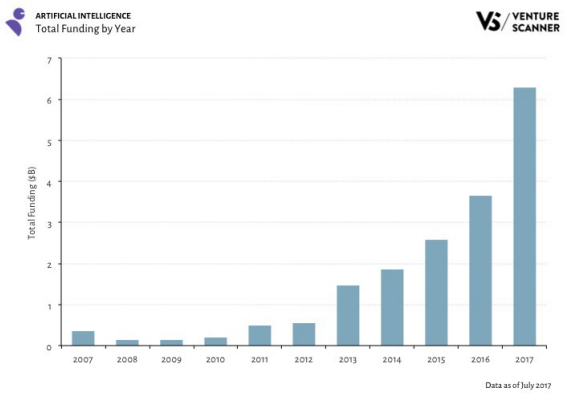


Figure 2.5: Total Funding By Year

The above graph summarizes the total funding of each year. Similarly, the job market requiring AI skills, according to [11], there is a noticeable rapid growth in AI Job requirements in 2017-2018.

2.2 Neural Networks

A neural network is a network, circuit of neurons or a web of nodes connected to each other. It is a system which is modelled on the human brain and nervous system. Or it is a series of algorithms that endeavors to recognize or detect the relationship in a set of data through a process that mimics the way a human brain operates.

Our brain is a very complex neural network. It has 10^{11} neurons and each neuron is connected to 10^4 neurons. These neurons use electrical pulses to communicate with each other. Below is the simplified model of a neuron taken from [31].

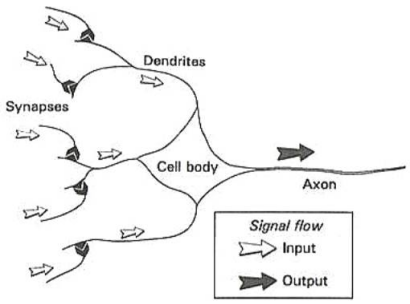


Figure 2.6: Highly simplified working of a biological neuron

As it can be seen from above diagram **"Axon"** is a output which is decided by the **"Cell Body"** of neuron. The inputs here are countable but in reality they are in thousands. This is how basically a neural network works.

2.2.1 Structure

The structure of artificial neural network works by creating connections between many different nodes, each analogous to single neuron in a biological brain. [12] The neurons can be physical or programmed in a digital computer. Each of the neuron takes many data inputs and then based on weights of the architecture, which produces an output that passes to other neurons in the architecture.

The nodes or neurons are tightly connected and organized into different layers. The input layers take the input, and output layer produces the result from the input. A Neural Network Architecture may have one or more hidden layers between the input and output layer, depending on the specific problem. A simple neural network is shown in the figure below.

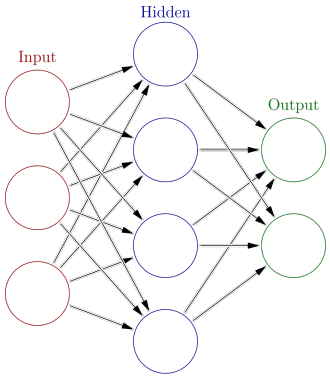


Figure 2.7: A NEURAL NETWORK

2.2.2 How Neural Network Learn

Typically a neural network starts with randomized **Weights** for all their neurons or nodes. **Weight** is the term typically used in artificial and biological neural network research, which is considered as a parameter which represents the strength of units between nodes. In simple words, a weight decides the influence of one node to the other connected node".

There are two methods for training a Neural Network. These are as follows. [12]

- **Self-Organizing ANN:** A self-organizing artificial neural network or **"Kohonen"** as it is known, is a network which tends to detect or recognize the pattern and relationship in the data. Researchers mostly use this type of network to analyze the data.
- **Back-Propagation ANN:** The Back-Propagation Artificial Neural Network is a type which is often used for cognitive research and for solving the specific problems in an application. It is trained by humans. During the training period, the spectator evaluates the results. If the result is correct the weights that produces the output are fortified else it is diminished.

2.2.3 Applications and Timeline of Neural Networks

Neural Networks have proved to be very useful in solving variety of real-world problems. The first of these applications that applied to the real-world problems were **ADALINE** and **MADALINE** (Multiple Adaptive Linear Elements) in 1959 by **Bernard Widrow** and **Marcian Hoff** [1]. Adaline was developed to predict the streaming bits from the phone line. Madaline was the first application that uses adaptive filters to remove echoes on phone lines. This Neural Network is still in commercial use in air traffic control systems. Similarly before Adaline, In 1957 **Frank Rossenblatt** introduces a single layer neural network, which was named

as **Perceptron** [44]. Perceptron is an algorithm for supervised learning of binary classifiers,to classify the given input data.

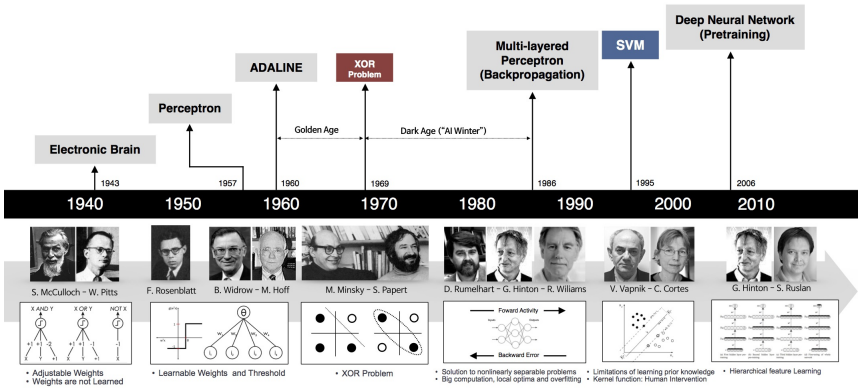


Figure 2.8: Neural Network timeline

The figure above shows the timeline of Neural Networks from 1940 till 2010.

As it can be seen that, it was all started from **S. McCulloch** and **W. Pitts's "Electronic Brain"**, by the time after AI winter, in 2006 G.Hinton and S.Ruslan came up with the idea of DNN which are commonly known as Deep Neural Networks, and is one of the most powerful and widely used algorithm till this day. The Neural Networks are based on three major factors

- Powerful Hardware
- Huge Datasets
- Powerful and Highlevel Libraries

Artificial Neural Networks are widely used in robotic factories such as diagnose malfunctions, adjustment of temperature, and controlling other machinery.

Convolutional Neural Networks or CNN are the most popular nowadays. Especially, in the field of Image Recognition. This type of Neural Networks has been used in many of the most advanced applications of AI including facial recognition, text digitization, image classification, and NLP (Natural Language Processing) [6].

A basic architecture of a neural network for classifying a Dog from image data is given below [22].

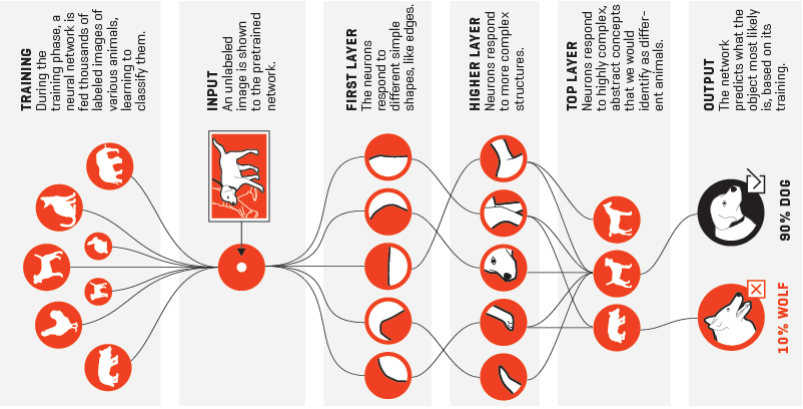


Figure 2.9: An example of Neural Network classifying a Dog

2.3 Machine Learning Fundamentals

Machine Learning is one of the hottest topic amongst students and researchers in the recent years. Machine Learning has been spread into our daily lives. Lets take an example of streaming services or most of the shopping sites. We all may have explored that when we see a video or a movie on a website, the application automatically begins to show you other movies of same genre, similarly when we buy or search for some specific things the applications begin to show us similar things, this is what we call Machine Learning. The application learn about the information that we like or want and then start giving us the suggestions. This section will briefly introduce Machine Learning, its History and it's kinds.

2.3.1 What is Machine Learning?

"Machine Learning", as the name suggests, is the computational process which provides machine an ability to learn independently based on experiences, analysis of patterns from a given data set without programming explicitly [26].

Let's take a simple example to define Machine Learning "To develop any kind of specific application, we usually have to program it or code it. This program or code is actually a set of instructions that we are telling the machine to follow. Whereas in machine learning, we input a data set to let the machine learn itself, by identifying specific features and analyzing the patterns in the given data set and learn to take decisions independently on its own."

To make it more understandable lets take an example, to classify whether a given image is of cat or dog. Lets assume we have a large amount of image data set of different cats and dogs. And lets assume both of the categories have 10 images in total.

Now for letting the machine learn we have to label the data as machine works in the form of numbers. So for each class of category i.e. cats and dogs we assigned a specific number which is known as the label of that specific category such as for cat - 0 and for dog - 1.

Now after selecting the learning architecture or model, we train our model so it can classify the whether the given image is dog or cat. This is how machine learning works to solve the specific problem.

The more data we use for training, the more better results we will get.

2.3.2 Brief History And Time line Of Machine Learning

Today, Machine Learning powerful tools are being a part of our daily lives, from self-driving cars to voice-activated assistants, image recognition, speech recognition and what not. The power of machine learning is immerse.

However, the idea of invention of machine learning have a long history. Recalling the previous section in which we discuss about famous research "Computing Machinery and Intelligence" by Allan Turing in which he asked: "Can Machines Think?"- a question with which we still wrestle with - [2].

Machine Learning, the term itself was coined by Arthur Samuel in 1959, an American Pioneer in computer gaming and Artificial Intelligence who created his first computer learning program in 1952, known as a **Game Of Checkers**. In the early days of Machine Learning, the researchers were already interested in having machines learn from data. The approach they attempted were mostly perceptrons and other models as discussed above, which later found as a reinventions of GLM (Generalized Linear Models) - [41].

In 1990 Machine learning got it's recognition as a separate field and as a subset of Artificial Intelligence. A time line of Machine Learning which we found really interesting is explained by [26] is given below.

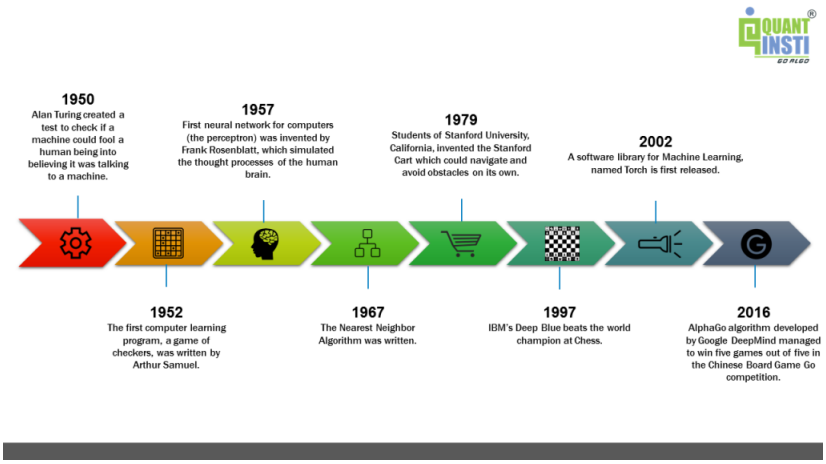


Figure 2.10: Brief history highlighting critical events in AI and ML

2.3.3 Types of learning Algorithms

The types of Machine Learning algorithms differ in their structure or working. Every algorithm have different type of input taking and outputting the result. These are as follows - [42].

- Supervised Learning
- Semi-supervised Learning
- Unsupervised Learning
- Reinforcement Learning

1. **Supervised Learning Learning:** Supervised learning algorithms are the task and techniques which have both data and labels. The data is known as training data and label is the tag associated with some specific output or final choice such as dog, cat, rock, plane etc. Supervised learning algorithms can be further classified as:

   - Classification: Classification is a type of supervised learning algorithm in which the machine tries to identify in which set of categories a new observation belongs to. Taking an example of this thesis, the model identify in which category a car falls into, when provided a new car as an input. Or we can take an example of assigning an email to check whether is a spam or not, from the trained model. Classification algorithms can be categorized as follows [29].
     (a) Logistic Regression:
     (b) K-Nearest Neighbour:
     (c) Support Vector Machine:
     (d) Naive Bayes:
     (e) Decision Tree Classification:
   - Regression: Regression analysis is the prediction that estimates the relationship from the variables. Regression predicts a numerical value based on previous observed data. It is widely used in forecasting. In simple words, regression analysis is the task of approximating from input variables to continuous outputs. For example approximating the price of the car which can be between 3000Euro-5000Euro.

2. **Semi-supervised Learning:** Semi-supervised learning is the technique which uses the unlabeled data together with labeled data while learning. In semi-supervised learning a small amount of data is labeled while a large amount isn't. Semi-supervised Learning technique falls under the category of Supervised and Unsupervised Learning.

3. **Unsupervised Learning:** Opposite to Supervised Learning, Unsupervised learning algorithm is presented with unlabeled data. It only takes the input data, and find structure within the given data. It is also known as Hebbian learning meaning learning without a teacher. Unsupervised algorithm learn from testing data that is not categorized or labeled. As Yan Lecun in his paper [16], director of AI research discussed about unsupervised learning - "teaching machines to learn for themselves without having to be explicitly told if everything they do is right or wrong- is the key to **true AI**". Most common unsupervised algorithms are clustering, anomaly detection etc.

4. **Reinforcement Learning:** Reinforcement learning is an algorithm in which the decisions are made sequentially. A good example would be a chess game. In simple words, in reinforcement learning the input depends on the output of previous input and so on.

2.4 What are DNNs?

DNN or Deep Neural Network is a neural network having more than two layers, which have a unequivocal level of complexity. Deep Neural Networks uses a complex mathematical equations to process data in complex ways [35].

Deep Learning as a subset of Machine Learning, is a neural network which uses a combination of layers of algorithm to process data and to understand human speech, or visually recognize objects. As mentioned above, the first layer of the network is known as **input layer**, and the last layer is known as **output layer** which produces the result after processing the given input data. The layers in-between input and output layer are known as **hidden layer** which are used depending on specific problem. Each layer is an algorithm which contains an activation function. An **activation function** is a function which is used to perform the linear or non-linear regression, depending on the task or problem.

Deep learning networks are known and distinguished from other neural networks because of their depth, i.e. number of node layers between input layer and output layer through which the data passed.

Earlier types of neural networks that were described above such as Perceptrons have only one input layer and one output layer, and at-most one hidden layer in between. A neural network having more than three hidden layer is said to be "deep" learning network or deep neural network.

In deep learning or deep neural networks, each layer of nodes learns or trains on the set of features that are provided by previous layer of nodes. Getting deeper and deeper, the deep neural network manage to train itself on the key features of given data set and passes it to the next layer for further feature extraction. This feature extraction of data is known as feature hierarchy.

This gives DNN an ability to perform and to handle large amount of high-dimensional data sets with millions of parameters. [39]

2.4.1 DNNs in general

As discussed above about DNN's training itself on huge amount of high-dimensional data. During early era of neural networks one of the complex and difficult problem was to handle handling unlabeled or unstructured data. Unstructured data is a raw media such as text, picture, videos or audio recordings. Therefore, after the invention of Deep Neural Networks, it was considered as the best and most reliable neural network in handling large amount of unlabeled or unstructured data.

For instance, a DNN can take a millions of images and with the help of nodes and feature extraction it begin to cluster them with the similar at each corner. Lets take an example of simple email system labeled data in a customer service portal. Lets assume we have three types of classes within the given email data. i.e.

1. Angry customers emails who are not satisfied from the services provided.

2. Satisfied customers emails.

3. Spam emails.

Assuming as an input data set to the Deep neural network, the network will try to train itself by clustering the given input data set into three vector corner spaces. Such as, angry emails of customer on one corner and spam and satisfied emails to the other corner. Same applies to the voice messages.

A convenient way that deep neural networks provide is that these networks perform feature extraction without intervention of humans, unlike many machine learning algorithms. When training on unlabeled data, each layer of nodes in DNN repeatedly learn the key features of the given input data and try to reconstruct it by samples of features. **"Restricted Boltzmann Machine"** a generative stochastic ANN do the so called reconstruction in this manner. In this process, the neural network learns to recognize the relations between the relevant features and provides the results.

For having a higher performance, the deep learning neural network which was trained on labeled data can then applied on raw data or unstructured data, the more inputs a model trained on, the more accurate results can be obtained. The algorithms which are said to be bad can even be more useful and can outperform the good algorithms if they are trained on more data than good algorithms.

Deep Neural Network ends in similar output as every neural network, but has a logistic or a softmax is a function that assigns the label to the output. We call it prediction in a broader sense. Given a raw image of a car, based on the specific problem, a DNN can decide that the given input is 90% car, if trained correctly.

An interesting example of How DNN sees an image can be seen below. Image is taken from [17].

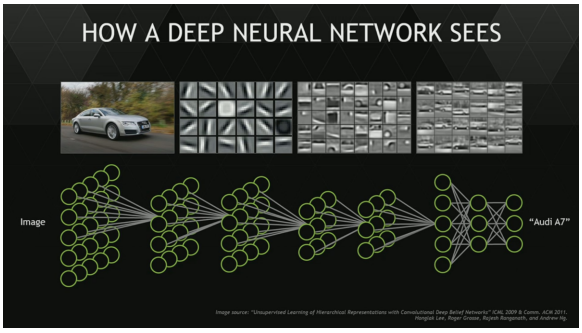


Figure 2.11: How a deep neural network sees an input

2.4.2 Application Work Flow Of Deep Learning

The application work flow of deep learning or machine learning are same. Below are the 7 steps of application work flow of both learning methods, that are important to reach the required goal, also for this particular thesis.

1. **Gathering of right data:** The first step of every learning method is to gather the data for specific problem. For instance, for the classification of the image whether it is a dog or cat, we have to gather different images of cats and dogs. Nowadays, the data sets can be downloaded from internet, where a lots of online communities such as Google's Kaggle which allow users to find or publish high-quality data sets for machine or deep learning purposes.

2. **Data preparation:** After gathering the required data set, we have to prepare it for training purposes. For preparation, the data set should be split into three parts [43] i.e.

   (a) Training: Training data sets are the sets which are used for training or fit into the model.

   (b) Validation: The validation data sets are used to tune the hyper-parameters i.e. the architecture. In order to avoid over-fitting, it is compulsory to have validation data set along with Training and Testing.

   (c) Testing: The testing data set is use to test the results of the model, i.e. by giving an image as an input to test how the model reacts or output's the certain input.

   after splitting, the image data set should then be converted into machine readable form, for example conversion of images in numpy format(python) or in any other standard format.

3. **Pre-processing Of image data set:** In case of image data set, it is a good practice to pre-process the image data set, the pre-processing comprises of many image processing methods, such as normalization, resizing etc.

4. **Choosing the learning model:** After preparing the data and pre-processing of images, we are ready to train our model with the given image data set. Over the recent years, as discussed above in the rise and advancement of AI into subsets, scientists and researchers have published and created numerous models or architectures for Machine or Deep learning methods, such as AlexNet, Inceptionv3, VGG19 and many more.

5. **Evaluation:** When the training of the chosen model done, we can now evaluate the results by testing the model, giving testing data set as an input to check the performance of the model and whether the obtained output is correct or not.

6. **Changing of parameters Or changing of model (optional):** After the evaluation of results, the model can be re-trained by changing the hyper-parameters such as changing weights, or by changing the size of each hidden layer to get more better performance and efficiency of the model.

7. **Result:** Having the best performance and efficient model, we can now use the model as an application for the predictions or obtain the required output.

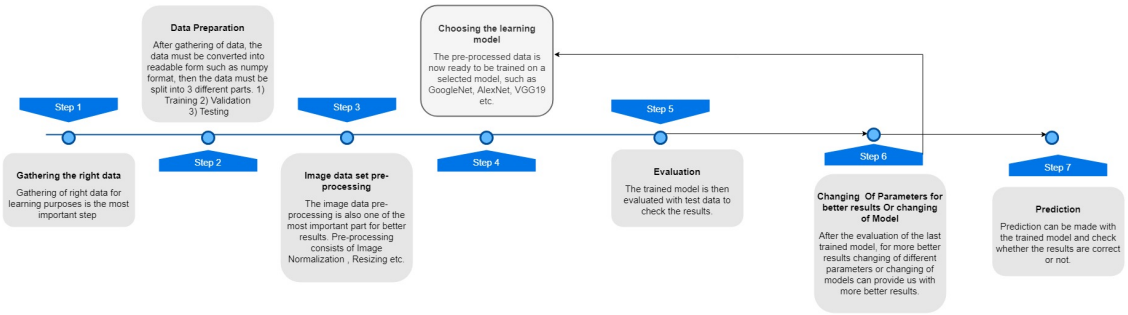


Figure 2.12: An example of Deep Learning Application Work flow

2.5 The CNN Network

The CNN network also known as Convolutional Neural Network are the class of Deep Neural Networks. The convolutional was taken from a latin word convolvere meaning "to convolve" or to roll together. The CNN network is most famous and most commonly used in analyzing visual imagery. CNN makes the explicit assumption that the data given are images, that allows us certain properties into architecture. They have applications in NLP, Image recognition, recommender systems, video recognition, image classification and medical image analysis [40]. They are also known as **"Shift Invariant or Space Invariant Artificial Neural Networks"**. Convolutional Neural Network is the most important part for this thesis as it is the network that has been used to support the completion of this particular thesis.

2.5.1 Brief History Of CNN

As Convolutional Neural Networks are considered very powerful network nowadays. It has a long history. In this section, we will discuss a brief history of CNN's and about their origin.

- "Neocognitron" also known as the origin of CNN architectures was invented by a japense computer scientist Kunihiko Fukushima in 1980. Inspired by the work of Hubel and Wiesel who showed the receptive fields in the visual cortex in 1950s and 1960s. Fukushima introduces Neocognition which has two basic types of layers in Convolutional Neural Network i.e.
  - **Convolutional Layers** which contains units whose receptive fields (A **receptive field** is a biological term which is known as a individual sensory neuron which is located in sensory space. e.g. the body surface , in which a stimulus will modify the firing of that particular neuron) cover the patch of the previous convolutional layer. The weight vector of these units are known as filters, that can be shared between the units.
  - **Down Sampling** is a layer which contains all the patches of previous convolutional layer that are covered by receptive fields.
- "Cresceptron" was introduced by J. Weng et al. which was similar to Fukushimas's neocognitron, but instead of using spatial averaging, J.Weng uses a method known as max-pooling. Max-pooling is a method in a downsampling layer which computes maximum activations of the units in its patch. Max-pooling is used in modern CNN architectures [40].

- TDNN or Time delay neural networks are said to be the first convolutional network introduced by Alex Waibel in 1987. Waibel's TDNN achieved shift invariance, by utilizing weight sharing using backpropagation training. Today, TDNN are known to be the best performance in far distance speech recognition.

- Image Recognition with CNNs: In 1989 Yann Lecun used back-propogation to learn the convolutional kernel from images of hand-written numbers. This automatic learning produced more better results and performed better than manual learning techniques. Thus, CNN became a foundation of modern computer vision especially image recognition and classification problems.

- LeNet-5: In 1998 a french computer scientist working primarily in the field of Machine Learning and Computer vision, Yann LeCun, came up with an idea of LeNet-5, a pioneering 7-level convolutional network that helps classifying digits in images. LeNet-5 architecture was applied on several bank hand-written checks to recognize numbers but on 32x32 pixel images, because the ability to apply the same architecture on high resolution images require more computing resources.

- SIANN: SIANN or Shift Invariant Artificial Neural Network was designed by W. Zhang et al. [46] in 1988. In this neural network the author proposed an architecture based on CNN for image character recognition, that was later modified for medical image processing and applied on detection of breast cancer in 1991.

Although, the invention of CNNs were in 1980s but the real breakthrough starts in 2000s when more computing power was begin to require for high-resolution images. As mentioned above about LeNet-5 which was constrained by the availabilty of computing resources.

In 2004, K. S. Oh and K. Jung showed that the performance and working of neural networks can be greatly accelerated by using GPUs. In their implementation they concluded that using neural networks with GPUs for computing can be 20 times more faster than the standard CPUs.

Similarly, years after years scientist proved that using GPUs with neural networks even with deep neural networks with many layers can be efficient and better in performance and results than CPU.

In 2012 an ImageNet Challenge winner GPU-based CNN was introduced by Alex Krizhevsky [14], which is a very deep CNN with more than 100 layers named as "AlexNet".

2.5.2 Architectural Design of Convolutional Neural Networks

As described above CNN's is neural network that consists an input and output with multiple layers in between known as hidden layers. These hidden layers are comprised of convolutional layers, a RELU layer, pooling layers, FC layers i.e. fully

connected layers and normalization layers. A simple architectural design of convolutional network is a s follows [40].

- **Convolutional** layers apply convolution to the inputs and passes the result to next layer. A convolution is a mathematical algorithm which is applied on two operations, assuming f and g to produce a third operation that expresses How the shape is modified by other. Convolution has many applications including probability, computer vision, signal and image processing, differential equation etc. Each convolutional node or neuron process the data based on its receptive fields. A fully connected layer is used to learn features but this type of layer is not practical to use on images, for example a 100 x 100 image has 10000 weights for each neuron in the second layer, that makes it really huge when the network gets deeper. Convolutionals play quite a big role in this kind of problems by reducing the number of free parameters, that allows the network to pass fewer parameters to the next layer.

- **Pooling** Pooling is the layer that reduces the dimensions of the data. Pooling layers take the outputs of all the last layer and combine them into one layer and then passes it to single neuron to the next layer. There are two types of pooling layers: Local Pooling and Global Pooling layers. Local Pooling applied on small clusters mostly 2 x 2, and Global Pooling applied on the whole convolutional layer. Pooling can also compute maximum value and average value from each of a cluster known as Max Pooling and Average Pooling respectively.

- **Fully Connected or FC** is a layer that connects the node of one layer to the node of another layer. The work of fully connected layer is same like multi-layer perceptron (MLP).

- **Receptive Field** As in every neural network, every node recieves a kind of input from its previous layer of nodes. In convolutional layers nodes or neurons receives input from a specific subarea of previous layers of nodes, these areas are 2D in shape e.g. 5 by 5. The input area of this node is known as receptive field of that neuron. So, in FC a receptive field in entire previous layer.

- **Weights** is a basic unit of a neural network. The function on the input is determine by a vector of weights and bias. It is a set that allows nodes to provide related outputs. If a neuron takes 4 inputs, then the no. of weight values are 4 also. The activation function of a neuron is determine by the product of weight and input plus bias. i.e.

$$output = activation((weights * input) + bias)$$

A basic convolutional neural network architecture can be seen below.

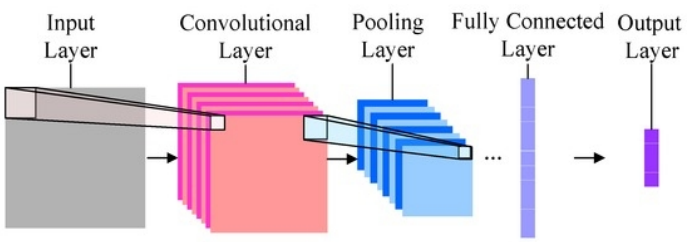


Figure 2.13: A basic structure of Convolutional Neural Networks

2.5.3 Getting Deeper with Convolutional Neural Networks

CNNs or Convolutional Neural Networks are Deep learning networks that are famous in classifying images, cluster the images by similarity and recognize objects in the images. The CNN algorithms are the algorithms that can classify or recognize faces, humans, street signs, tumors and many other aspects of visual data. CNN can also perform well if given sound as input represented visually by spectrogram.

The efficiency in image recognition is one of the main reason that the technical world is taking interest in convolutional neural networks. CNNs are helping boosting up major advancements in the field of image processing, the obvious real world applications are self-driving cars, robotics, medical diagnoses and much more. [39]

Like every neural network convolution neural network does not take images as humans do. They perceive images as volumes i.e. 3D - 3 dimensional objects instead of flat width and height. So a CNN takes a colored input image as a squared box whose width and height is decided by the number of pixels in that particular image, and the depth of that image is three layers deep i.e. Red, Green and Blue, referred as RGB or channels.

An input image is describe mathematically as matrices in the form of 224x224x3. On each of three layers the dimensions changes as the intensity of R, G and B changes which are expressed in numbers and that particular number will be an element in one of the three, two dimensional matrices, which combine themselves to form a image volume.

The particular image is then pass to convolutional network which helps identify those signals that are significant enough and helps CNNs to classify images. Instead of taking one pixel at a time, the CNN is used to take the patches of images in a matrix form which passed them through a filter. A filter is also a square matrix same as the patch but smaller than the image itself. It is also called as kernel. The filter is use to find the patterns in the given pixels of the image. A demonstration can be seen below.

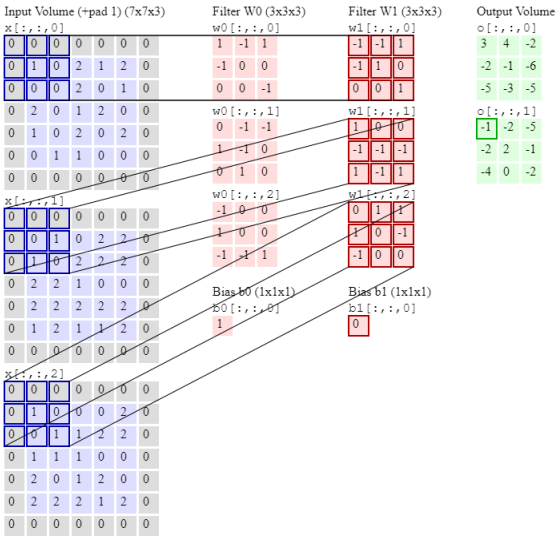


Figure 2.14: A basic architecture of how CNN applies the filter to image

2.5.4 How do CNN works?

As we discussed above how CNN works, in this section we will explain, which principle methods a CNN take to identify an image. For every CNN there are four main operations that are mentioned above in Figure 2.13.

As mentioned earlier, a computer sees an image as a matrix of pixel values. A coloured image have three channels i.e. RGB, stacked over each other having pixel values range from 0-255. And the gray scale image has just one channel, and each pixel value in the matrix range from 0 - 255, where 0 is black and 255 is white.

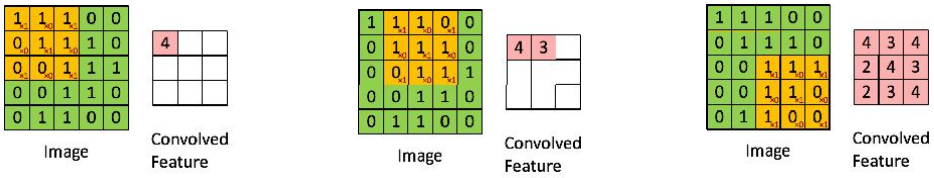
The four main operations with simple explanation are given below [36].

- **Convolution:** The main purpose of a convolutional layer is to detect or extract key features of the image. These features are extracted by filters or squared box for more simplicity.

  As we said, a computer take an image as a matrix of pixel values. Assume a 5x5 and 3x3 matrix given below whose pixel value ranges from 0-1.

(a) 5x5 Matrix ranges from 0 to 1 (b) 3x3 Matrix ranges from 0 to 1

The convolution of Fig (a) i.e. 5x5 and Fig (b) i.e. 3x3 above can be computed as shown below.



(a) Applying convolution (b) First stride (c) Last stride

The figure above demonstrates how a convolution works on 5x5 matrix and 3x3 matrix, for each step the 3x3 matrix is sliding incrementally by 1 which is also known strides, and applying the element wise dot product and then adding the outputs to get the final value. Also the important thing to notice that 3x3 matrix sees only a part of 5x5 matrix in each stride.

In Convolutional Networks, the 3x3 matrix is known as a filter or a kernel or more simply a specific feature detector from an image by sliding and compute dot product. The final matrix which is formed as a result [see Fig (c) above] is known as a feature or an activation map. Another thing which is evident

in the above figure is that different values of filter will return different feature or activation maps. These different values can be used for various operations, such as identity, edge detection, sharping of image, blurring or normalization etc. This is how a CNN works and train itself on an image by sliding down and calculating the key features of that particular image. So, it is concluded that more number of filters we have, the more features will be detected and better will be the performance of our network.

The size of feature map or activation map is controlled by three important parameters.

- Depth: In Convolutional Neural Network, the depth is the number of filters used in the convolution operation. For instance using three different filters on an image will have a depth of 3, similarly using 5 filter will be taken as depth of 5 and so on.
- Stride: Stride is the number of step or incremental moving to slide the filter over image. If the stride is 1 the filter will move one pixel at a time. Similarly, if the stride is 3 the filter will move three pixel each time.
- Zero-Padding: Using zero-padding will apply 0's to the edges of input image. Zero-padding parameter allows the developers to control the size of feature maps. It is also known as wide convolution, and the feature map without zero-padding parameter is known narrow convolution. A simple example can be seen below.

Figure 2.17: Zero-padding applied to 5x5 matrix

- **RELU Layer:** is an activation function which is commonly used in neural networks, especially in CNNs. RELU stands for Rectified Linear Unit which is used to remove negative pixel values after convolutional layer. Since non-linear image data is really useful in CNNs to train itself more better, and to learn features more efficiently. It is defined as below:

$$Output = max(0, Input)$$

A graph explaining RELU can be seen below:

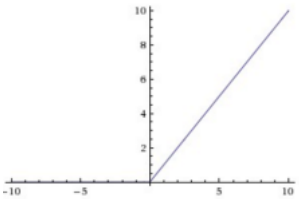


Figure 2.18: Graph Representation Of RELU [36]

As it can be seen from above equation that, RELU is a function that replaces negative values to 0. A basic example of a rectified input image and the output after passing it through RELU activation function is given below.

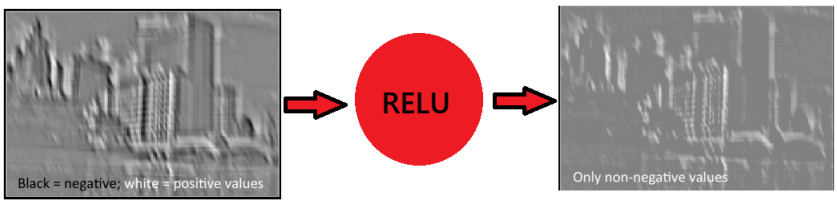


Figure 2.19: An rectified input image passes through RELU layer

- **Pooling or Sub Sampling:** The pooling step includes spatial pooling also called downsampling function which reduce the dimentions of every feature or activation map but does not loose important information of input data.

The spatial pooling consists of Max, Average, Sum etc. In case of max pooling, the filter takes the maximum element from the features map. Max pooling can be defined as:

$$Output = max(InputImageFilter)$$

Pooling identifies the most important feature in the square box that is known as filter or it can be defined as a feature extractor matrix. Similarly in case of Average and Sum pooling we can take the output as average of the sample or the sum of all values in the sample, depends upon the specific problem. A visual example of the a input matrix and the output of max pooling can be seen in the figure below.

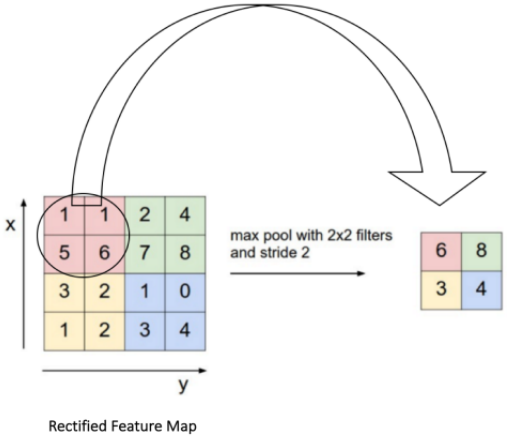


Figure 2.20: Maxpooling of a input matrix [36]

Pooling helps in making the input data smaller, reduces the number of parameters which effects on the computational power of a network, and helps in making the input data transformed and distorted.

- **Classification or Fully Connected Layer:** As discussed above, the fully connected layer is a kind of multi layered perceptron that uses a softmax activation function the final output of the layer. The term FC is defined as every neuron or node of the previous layer should be connected to every other node of the next layer.

  Using fully connected layers in the CNN helps in classifying the input data based on the learning of classes while training. The output from convolutional and output of RELU function after it, provides high-level representation of important features from the input image. Which can be then passes through FC layer for classification purposes. The output probabilities of an FC layer is always 1, that is because of the activation function i.e. Softmax, which takes the arbitrary values as an input and output a vector of values between 1 and 0 that sum to one.

Check Intesars report and other saved material for reference

2.5.5 Used CNN Architectures

As mentioned earlier, CNNs, ConvNets or Convolutional Neural Networks are the kind of multi-layered neural networks, that are mainly used for classification and recognize visual patterns from pixel of image. Today, the researchers have proposed numerous amount of different CNN architectures, using hidden layers for classification.

In this section we will discuss about the CNN architectures that we used in this thesis. Before, moving on, we are going to discuss a bit about ImageNet is and how CNN architectures are being rated as the top architecture.

"ImageNet" is a database which is designed to store huge amount of image data, to use in visual object recognition software. It is an image database organized according to world hierarchy, which is consists of nodes, each node stores hundreds and thousands of images. Till to date, ImageNet have stored more than 14 million images, with 1 million of images whose bounding boxes are also provided.

This "ImageNet" database providers also holds an annual software competition known as "ImageNet Large Scale Visual Recognition Challenge" which is abbreviated as ILSVRC, the aim of this contest is to provide the best architecture for classification problems or other image processing/computer vision problems [3].

The following figure below shows top architectures proposed in ImgeNet software contest and there image classification error.

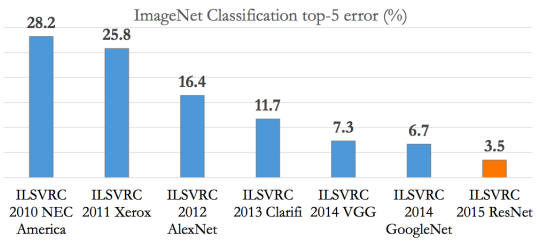


Figure 2.21: ImageNet CLassification error – CNN architecures

In this section as mentioned above, we are going to elaborate the CNN architecture that support this thesis problem. As mentioned earlier in Chapter-1, this thesis is based on CompCars dataset, an already researched problem in which the researchers used two of the powerful CNN architectures, so in this thesis the architectures we used are AlexNet, GoogleNet-Inception, and VGG19 that are explained below.

AlexNet

AlexNet is a convolutional neural network architecture who won the ImageNet competition in 2012 with the classification error rate of 16.4% less than the Xerox (25.8%) who won ILSVRC in 2011 and NEC America (28.2%) who won the competition in 2010. The paper was published namely – "ImageNet classification with Deep Learning Neural Networks [14]" by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton.

The architecture of Alexnet was much complex and larger than previous CNNs before. It has 65 million parameters including 650,000 neurons and it was recorded that the architecture took 5 to 6 days to train on two GTX 580 3GB GPUs. However, today there are algorithms that are much more complex than AlexNet, but back than in 2012 AlexNet architecture was considered to be really huge. The input image for AlexNet architecture is required to be of 256x256 in size. That means all the images in training set or testing set needs to be 256x256 size. After inputting an image of 256x256, random crops are generated of size 227x227 for feeding into the first layer of AlexNet. An important point which is to be noted here that the paper claims that the instead of 227x227 the network inputs to be 224x224 which is a mistake as claimed by many researchers and scientists [23] and [33].

AlexNet is a deep convolutional neural network that is same as LeNet by Yann Lecun as discussed above, but AlexNet consists of more filters on each layer. It consists of 5 layers of CNN and 3 FC(Fully Connected) layers. A short architectural working of AlexNet are given below [23].

- The first layer in AlexNet consists of 96 kernels of size 11x11x3 with stride 4, then 5x5 and then 3x3.
- The first two layers are then followed by Max Pooling layers.
- After first two layers that are connected by Max Pooling layers in between, the 3rd, 4th and 5th layers in AlexNet are directly connected to each other, can be seen in the figure below.
- The output of 5th layer is overlapped by an other Max Pooling layer while reaching to the series of two Fully Connected dense layers.

- The last dense layer is then send the final output to the softmax classifier with 1000 class labels.
- A ReLU nonlinear function has been applied after each layer in AlexNet.

The architectural diagram of AlexNet can be seen below.

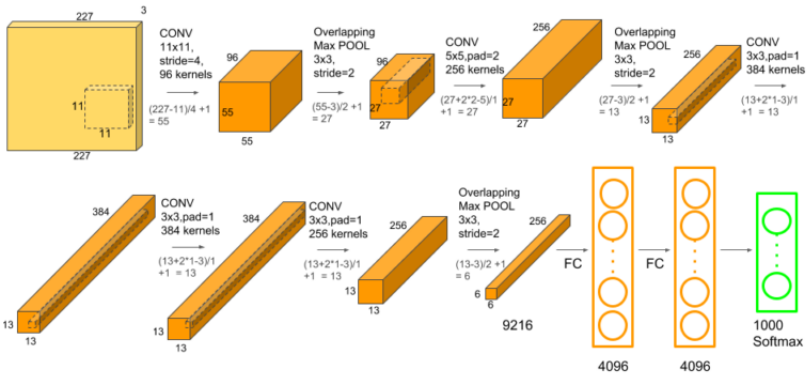


Figure 2.22: AlexNet Architecture

Reducing Overfitting

What is Overfitting? Overfitting is one of the general problem which occurs when training the models. Lets take a simple example to explain what overfitting really is.

"We all may have seen or met a student who performs very best during the lectures, but get the least grades or sometimes failed when it comes to exams or the tricky questions, did you ever wonder why? This is mainly because he only memorized the answers covered in the lectures without understanding the main concept of the topic." Same is what over-fitting is in terms of training of a model. The model will train it self well on training data with a good or may be the best accuracy but eventually it will fail when the unseen testing data is given as input. This is due to the same reason as the example above, that the model does not understand the real underlying concept of the data.

The authors of alexNet used different methods for reducing the over-fitting of the model. These are as follows.

- **Data Augmentation**: Giving inputs of same image with various variation to the model can help prevent over-fitting. The advantage of Data Augmentation is that you're forcing your model not to memorize only one training input. There are many different methods of data augmentation, the authors of alexNet uses two different kind of image augmentation methods i.e. Data Augmentation by Mirroring of image and Data Augmentation by generating random crops. Mirroring is a computer vision technique that convert or flip the given image horizontally. A simple example can be seen below.

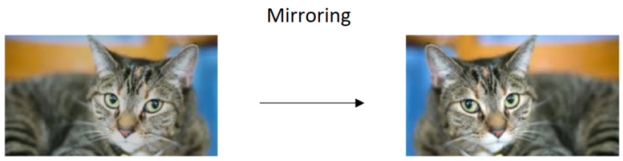


Figure 2.23: Mirroring – Image Augmentation [8]

Likewise, the authors also use another augmentation technique by generating the random crops. The random crops is the cropping of the image randomly from 256x256 in different size in to the size of 227x227. An example of random crops can be seen below.

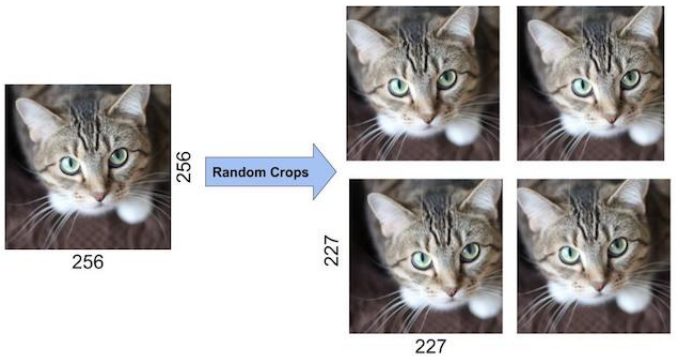


Figure 2.24: Randomly generated crops. [23]

The above figure seems to be same even after randomly cropping but it's not exactly the same. The shifting of pixels can be seen in all of the outputs from 256x256 to 227x227. This kind of technique teaches the model that shifting of pixels does not change the fact or the important information in the image.

- **Dropout**: Dropout is another technique for reducing over-fitting. It was introduced by G.E Hinton in the paper "Improving neural networks by preventing co-adaptation of feature detectors" [9]. It is a regularization technique, in which a neuron is dropped from the network. The randomly selected neurons are then ignored in training, and does not take part in either forward or back propagation. Dropout increases the number of iterations needed to converge by a factor of 2. It is claimed that without dropout AlexNet would overfit exponentially [23].

Today dropout is an important part of every convolutional neural network as it helps a lot in reducing over-fitting.

InceptionV3

Success of AlexNet made a boosting speed in the inventions of Convolutional Neural Networks. Deep Neural Networks are computationally expensive, for making it

cheaper, Inception was introduced to decrease the number of parameters and limiting the number of channels. The Inception architecture was developed in 2015 and introduced by GoogleNet [34] which was based on the thinking "We need to go more deeper". It has 7 million parameters with 42 layers. Inception architecture have different versions starting from V1-V4 till today.

Inception architecture is heavily engineered to decrease the time of training and increasing efficiency. Inception architecture consists of modules with different kernel sizes. The aim was to reduce the number of parameters in CNN that can help to achieve better results while using cheaper computational power. The authors replaces the number of filters used in older versions with less amount of filters making the architecture look wider than deeper. A simple example can be seen below.

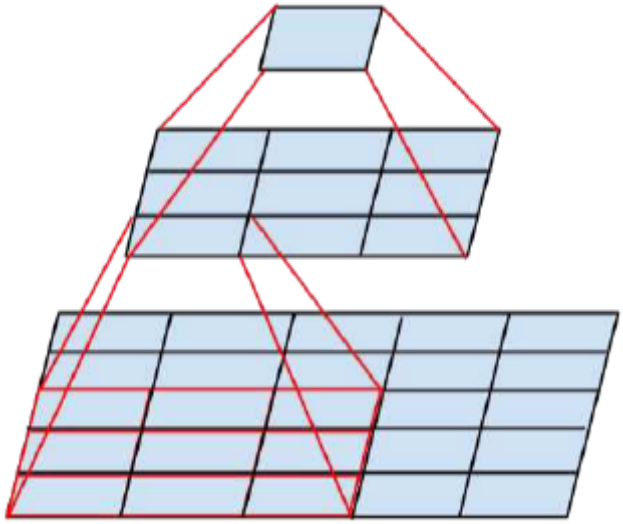


Figure 2.25: Two 3x3 Conv Layers replacing 5x5 [21]

As it can be seen from the above figure, A 5x5 filter of layer is replaced by 3x3 that is because 1 layer of $5 * 5 = 25 NumberOfParameters$, whereas by replacing 5x5 with two layers of 3x3 gives us $3 * 3 + 3 * 3 = 18 NumberOfParameters$ i.e. the number of parameters are reduced by 28%. The authors uses factorization method to reduce and apply the layers by replacing them from the old one. A single module or module A (Inception Module A – as in technical terms) can be seen below in the Figure 2.26.

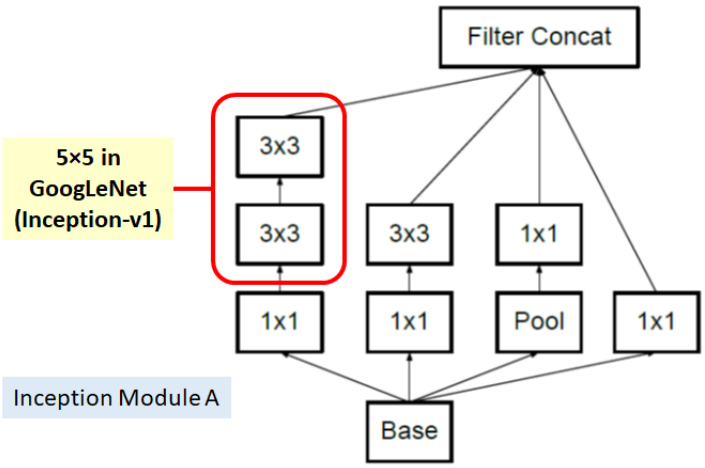


Figure 2.26: Inception Module A architecture

VGG19

VGG a.k.a Visual Geometry Group was invented by Karen Simonyan And Andrew Zisserman in 2014 [30]. The idea of VGG is same like Inception to have a deeper network. The authors of VGG introduces two different versions named as VGG16 and VGG19. The idea behind this invention was to increase the depth by using very small 3x3 convolutional filters in VGG16 and VGG 19.

The main difference in VGG16 and VGG19 is VGG16 consists of 16 deep layers while VGG19 consists of 19. Both architectures use two or three stacks of 3x3 filters with strides having a Relu Activation function and under max pooling of 2x2. Both the architectures have two fully connected layers on the top, having 4096 neurons or nodes followed by softmax classifier.

These architectures secured first and second position in ImageNet competitions held in 2014. For this thesis VGG19 is used and applied for evaluation.

2.6 Exploring Dataset

Data is one of the most important and crucial part in deep neural networks and consider as a backbone of every image classification learning problem. For solving a classification problem, it is necessary that the gathered data should be of high and good quality, which is capable of generalizing a model well enough. As mentioned earlier, for this particular thesis a data set taken from Comprehensive Car also known as CompCar is taken for solving the vehicle classification problem.

The data set provided by CompCar is enormous and contains thousands of images with multi-label data containing hundreds of makes and models with different viewpoints.

The following section of this chapter will further describe the nature of the data set which is used to support this thesis.

2.6.1 The CompCar Data set

The Comprehensive Car data set [45] is one of the huge data set containing hundreds of images of different car makes and models. This data set contains data from three scenarios including **"Web Nature", "Surveillance Nature" and "Vehicle Parts"**.

The Comp Car data set is prepared keeping in mind following problems:

- Car model Verification
- Fine Grained Classification
- Attribute Prediction

The **Web Nature** data contains more than 100 car makes with 1700 car models, having total of more than 100,000 images containing the entire car images and more than 27000 images containing the car parts. The following table below shows the exact figure of number of images each data set contains.

DataSet	Number Of Images
Web Nature	136,727
Survalliance Nature	50,000
Vehicle Parts	27,618

Table 2.1: Data Set kinds and Number Of Images

2.6.2 Web-Nature Dataset

Web Nature data set contains the images structured in the folders as "make_id/model_id/released_year/image.jpg". This kind of data set was collected from web resources including all the publicly available images from hosting sites, forums and from search engines especially Google. As mentioned above the web nature data set contains more than 100,000 images depicting the information of entire car from different view points. **For this thesis we only use Web Nature Data set as it provides the most detailed and useful information of cars. The data set is split into three different parts i.e. training data set with 1000 samples of each car make, training data set with 2000 samples of each car make and training data set with 4000 samples of each car make**. There are total of 163 car makes and 2004 car models provided by Comp Car in Web Nature data set. Some examples of Web Nature images are given below.

Figure 2.27: Web Nature Data

Each category in web nature data set includes different viewpoints of same car make or model, which can be seen in the figure above. The CompCar data set providers distribute the quantity of each labeled car images in different view points. The viewpoints are not balanced among different car models, as the images of some less popular car models are difficult to collect [45]. The table below shows the quantity distribution of labeled car images in different viewpoints.

Viewpoint	No. in total	No. per model
F (Front)	18431	10.9
R (Rear)	13513	8.0
S (Side)	23551	14.0
FS (Front-Side)	49301	29.2
RS (Rear-Side)	31150	18.5

Table 2.2: Quantity distribution of labeled car images of different view points

The 5 classes of viewpoint explains that the viewpoint is the angle or side of the car from where it is captured, which is shown in figure above.

2.6.3 Surveillance Nature Dataset

Another huge data set provided by CompCar is the images of different cars taken from surveillance camera hence it is named as Surveillance Nature of data. The total number of images in this data set is 50,000, however as mentioned above only web nature data is used in the completion of this thesis hence we are only going to discuss a bit about Surveillance Nature data to get a rough idea about the whole data set. The images captures from surveillance camera are mostly from front view that can be seen in the figure below.

Figure 2.28: An example of Surveillance Nature data set

As it can be seen in the above figure that the images provided in Surveillance Nature has a large appearance variations due to the varying conditions of light, weather and traffic etc.

2.6.4 Vehicle Parts

The vehicle parts data set as the name suggests are the images of parts of different cars provided by CompCars mainly for the use of attribute prediction problems. The vehicle parts data set contains eight car parts for each car model. These includes four exterior parts i.e. Headlights, Taillight, Fog-light, and air-intake and four interior parts i.e. gear lever, dashboard, console and steering wheel. The quantity distribution of the vehicle parts are given below.

Vehicle Part	No. in total	No. per model
Headlight	3705	2.2
Taillight	3563	2.1
Fog-light	3177	1.9
Air-Intake	3407	2.0
Console	3350	2.0
Steering Wheel	3503	2.1
Dashboard	3478	2.1
Gear lever	3435	2.1

Table 2.3: Quantity distribution of labeled car images of different view points

Some of the images of vehicle interior and exterior parts are given below.

Figure 2.29: Vehicle Parts

The above exploration of data set concludes that CompCar is a versatile and high quality data set which offers massive amount of meta-data attached with every image. Below are the labels attached with the data-set that can be use in solving various classification problems. These are

- Car Make
- Car Model
- Car Released Year
- Car View Point

An important thing to note here that each label of the car for example the viewpoints are denoted by numbers i.e. -1 (Uncertain), 1 (Front), 2 (Rear), 3 (Side), 4 (Front-Side), 5 (Read-Side). A part of the above labels described above, CompCars also offers more labels attached with every car model. These are bounding boxes and attributes. The bounding boxes of each image is given in the format 'x1 y1 x2 y2', where x1, x2 and y1, y2 are the width and height of the image respectively. Similarly the attributes of each image were given in the format model_id, maximum_speed, displacement, no. of doors, no. of seats and type.

The type refers to the type of the car, which are MPV, Minibus, Sedan, Crossover, SUV, Estate, Hatchback, Fastback, Pickup, Sports, Convertible, and Hardtop. The average resolution of each image in CompCar data set is 858.04 x 589.45 (w x h), however all the images used were resized according to the applied deep learning algorithms.

Chapter 3

Dataset Preparation and Image Processing

3.1 Dataset Prepration

As discussed earlier, data is the most important and one of the crucial part for deep learning environments. A good, clean and high quality data can give us a great leverage in solving learning problems efficiently. The data gathered for deep learning problems are mostly raw and it is a representation of human observation that have to be converted and prepared with different techniques before feeding it into the model.

Better results depends highly on the careful collection of data we feed in to the model. In image classification, it requires a lots of computer vision techniques for the cleaning, and shaping of the data as per CNN requirements. For this thesis,a significant amount of time was spent to structure, understand and converting the data in to machine readable format for the experiments.

This section will explain in detail how the data sets were prepared, in which scenarios it is split, and in which format the data set was converted. Furthermore, this section will also discuss about the preprocessing and augmentation techniques which we applied on our data set before feeding it into CNNs.

3.1.1 Preparing CompCar

For CompCars, as mentioned in section 2, only web nature data set is used for training and testing of the data, which consists of 136,726 images. This data set is really huge, keeping in mind limited amount of memory and computing power, the data was sub divided into three different sub sets. i.e.

- **Set-I:** Number of classes that have more than 1000 samples each.
- **Set-II:** Number of classes that have more than 2000 samples each.
- **Set-III:** Number of classes that have more than 4000 samples each.

The authors of Compcars also split the training data from Web Nature data set for their research. The following graph below explains the chunk of data taken from whole web nature data. As it can be seen from the graph above that for the training

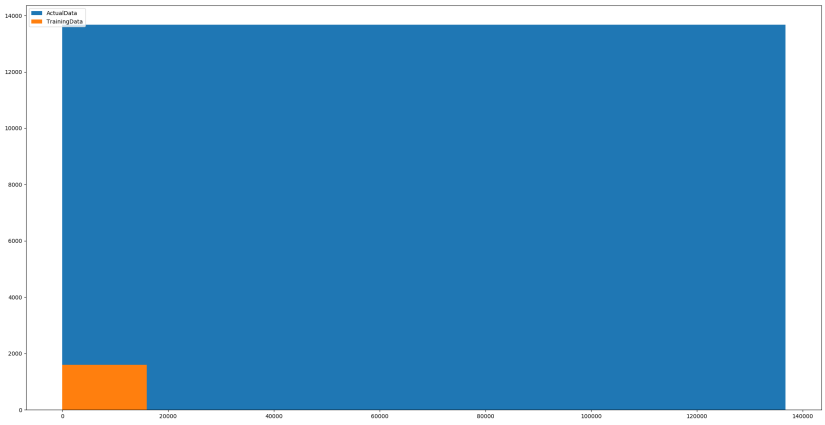


Figure 3.1: A distribution chunk of data taken from Web Nature Dataset in [45]

purpose a chunk of data which is 16016 out of 136,726 was taken by authors for the classification of CompCars.

3.1.2 Exploring subsets

As mentioned in above section, as the CompCar data set is really huge having more than 100,000 images that require powerful computational resources and efficiency. This huge data set is split into three different subsets namely Set-I, Set-II and Set-III.

In this section the different subsets will be discussed i.e. how many classes and images and what type of cars we have used in each subset for the classification.

- Subset-I: The first subset also known as Top-23 consists of the classes that have more than 1000 samples each. All the images across this subset are labelled with Car Makes, for instance BMW, Volkswagen, Mercedes, Audi etc. This subset comprises a total of 24,945 images. For training purpose this subset is further divided into testing (30%), training, and validation (30%) as per requirement of learning modules. The important thing which was kept in mind while further dividing this data set was that every split should have car images which belong to all 23 classes of the subset. Table below shows the exact number of images split into three sub categories i.e. training, testing and validation.

Type	Number Of Images
Total Images	33,842
Training	17,461
Testing	8,897
Validation	7,484

Table 3.1: Total No. of images used in subset-I

The following graph shows which car types are used in this particular subset and number of images from each car type in the training data.

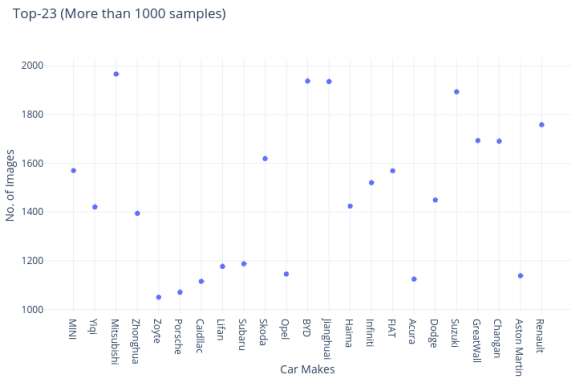


Figure 3.2: Top-23 Car Makes - No. of Images

- Subset-II: The subset-II also known as Top-13 consists of the classes that have more than 2000 samples each. Like subset-I, all the images across this subset are also labelled with Car Makes. Subset-II comprises a total of 14,213 images. For training purpose this subset is also divided into further subsets i.e. testing, training and validation. The testing data is 30% taken from the total of training data and and validation data is also 30% taken from the remaining training data.

The table below shows the exact figure of each split and total number of data taken for training of classifier.

Type	Number Of Images
Total Images	37,529
Training	20,543
Testing	8,181
Validation	8,805

Table 3.2: Total No. of images used in subset-II

The graph below shows the number of images as well as name of car types used in this subset.

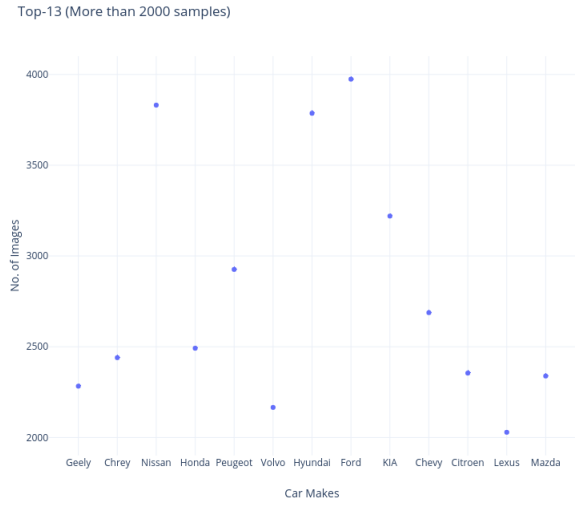


Figure 3.3: Top-13 Car Makes - No. of Images

- Subset-III: The subset-III is also known as Top-5 Car Makes, consists of the classes that have more than 4000 samples each. Same like subset-I and subset-II above, all the images are labelled with Car Makes, as the thesis focuses on the vehicle classification based on car manufacturers. The subset-III consists of total of 16,652 images, which is divided into different types same like other subsets above i.e. testing (30%), training(40%), and validation(30%). The table below shows the exact figure of each type and the total number of images.

Type	Number Of Images
Total Images	23,221
Training	11,656
Testing	6,569
Validation	4,996

Table 3.3: Total No. of images used in subset-III

The graph shows the number of images per class and the name of the car types from which category the image belong is given below.

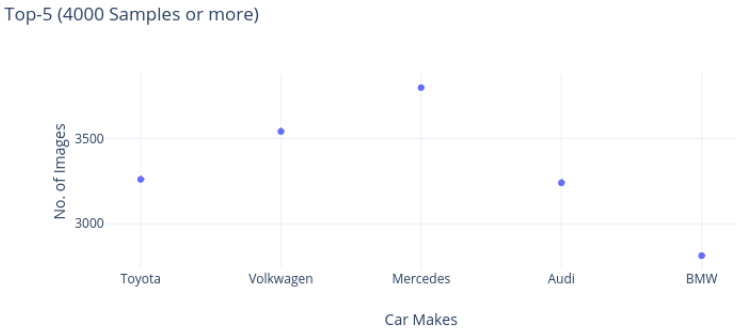


Figure 3.4: Top-5 Car Makes - No. of Images

The figure above shows that number of car images used from each class for training purpose after splitting 30% for testing data.

3.2 Image Preprocessing and Implementation

In terms of Computer Science, "Preprocessing refers to the data which is taken as input in a program whose output is taken as an input to another program". Image or data preprocessing is one of the important part in machine or deep learning. Preprocess data makes it easier and efficient for the learning models to learn and train themselves better and performs better in terms of results.

The following section of this chapter will discuss about preprocessing techniques that were used in this thesis as well as data augmentation techniques, and how and in which format the images were converted and normalized before passing it into CNN architectures, to avoid over-fitting of the model as explained in previous chapters.

3.2.1 Conversion Of Image Data Set in readable format

After gathering of CompCars data set and dividing into different subsets i.e. Number of classes having more 1000, 2000 and 4000 samples as mentioned above. The image data set has to be converted into a format which machine can understand and read.

As mentioned earlier, machine takes the input in form of number i.e. every input image is considered as 2D matrix having some values between 0-255 on each RGB layer. Even after dividing the data in different subsets, it wasn't easy to process all the images efficiently. A lots of problems have been faced, for instance Memory Exhausted problems, when reading all the images from each subset one by one. For this purpose Numpy library is used.

"Numpy" is a powerful python library that supports large collection of high level mathematical functions to operate on multi-dimensional arrays efficiently. For the purpose of converting image data into readable form all the matrices and values of each image was copied and pasted in numpy array of zeros while converting them into 'float32' mode after reading every image before resizing, normalizing , cropping of images with bounding boxes and augmenting them.

A sample of code explaining how the images were counted and then copied and pasted into the numpy array of zeros is given below.

```
# Counting Number Of Images From Subset
N = 0
for root, _ , files in os.walk(path):
    cdp = os.path.abspath(root)
    for f in files:
        name, ext = os.path.splitext(f)
        if ext == ".jpg":
            cip = os.path.join(cdp,f)
            N += 1
print(N)

# Initializing array of zeros -- Where N is the number of all images in a subset

IMG_SIZE = 224
imageX = np.zeros((N,IMG_SIZE,IMG_SIZE,3),dtype='float32')

# Reading Images and Copying and Pasting of image matrices into zeros array

i = 0
tr_labels = []
for root, _, files in os.walk(path):
  cdp = os.path.abspath(root)
  print(cdp)
  for f in files:
    ct = 0
    name,ext = os.path.splitext(f)
    if ext == ".jpg":
      cip = os.path.join(cdp,f)
      read = mpg.imread(cip)
      cipLabel = cip.replace('image','labels')
      cipLabel = cipLabel.replace('.jpg','.txt')
      nameL , extL = os.path.splitext(cipLabel)

# Reading x1,y1,x2,y2 co-ordinates for applying the bounding boxes to each image

      if extL == '.txt':
        boxes = open(cipLabel, 'r')
        for q in boxes:
           ct = ct + 1
           if ct == 3:
             try:
               x1 = int(q.rsplit(' ')[0])
               y1 = int(q.rsplit(' ')[1])
               x2 = int(q.rsplit(' ')[2])
               y2 = int(q.rsplit(' ')[3])
               readimage = read[y1:y2, x1:x2]
               resize = cv2.cv2.resize(readimage,(IMG_SIZE,IMG_SIZE))

               imageX[i] = resize


               tr_labels.append(cip.split('\\')[6])

             except Exception as e:
                print("Here",str(e))
      i += 1
      print(i)

print(len(tr_labels), len(imageX))
```

Listing 1: Conversion of Images into Numpy Format

3.2.2 Image Re-sizing

In this thesis, the main point of focus was to train the model that can recognize and classify that from which category or class the given input belongs to. It was considered that the images in our data set will contain exactly one car in the entire image input and the applied CNN can predict that only car whether it belongs to the certain trained data or not. However, there are some images that contains more than one car or the main car which is to be trained is not centered or cropped. This can lead us to the poor predictions from our applied CNNs.

For this purpose as it can be seen from the above sample of code, the bounding boxes provided by CompCars were used for each image to rightly cropping them and centered the most useful information of the image i.e. the main car which should be trained. The figure below gives a rough idea of how an original image is cropped by applying the bounding boxes.

Figure 3.5: Original Image (Up) and Cropped Image (Down)

In the figure above, it can be seen that how a most useful information i.e. an Audi in this case, was cropped and centered by using bounding box co-ordinates of that particular image. This kind of centered and main information is very useful when training on CNN . It will make more sense if we recall **Section 2.5.3 – Getting Deeper with CNNs**, where we explain how CNN work on an image.

3.2.3 Image Augmentation

Image Augmentation or data augmentation is a technique of creating new data with different orientations [37]. As discussed earlier, the augmentation techniques helps mainly in reducing over-fitting of the model. For this thesis, Keras Library [13] is used for the augmentation techniques. In the following section of this chapter, different techniques will be explained that has been used for augmenting CompCar data set for this thesis.

Once the images were cropped using bounding boxes as shown in the figure below, each image were resized with same width and height as per the requirement of CNNs i.e. 224 x 224.

3.2.4 Rescaling or Normalization

Colour images are in the form of RGB, different values of R(Red) layer, G(Green) layer, and B(Blue) layer combine together to form a coloured image. Since RGB channel is of 8 bit i.e. 2^8=256, that means each RGB channel has a range between 0-255.

This range is really huge for the computation, since CNNs uses complex matrix multiplication techniques, the range of each image was changed from 0-255 to 0-1 by dividing each pixel of the image by 255. This process is known as Normalization of an image.

3.2.5 Image Rotation

Image rotation is a technique which takes an image as input and then rotate it in different angle that is provided. In the figure below an input image(left) is rotated to 40 degrees.

Figure 3.6: Image Rotation Technique

As it can be seen from above rotation technique example, how an input image(left) is rotated to 40 degrees in every iteration.

3.2.6 Width Shift Range

The width shift range is a technique which apply horizontal shifts to the particular input according to the value given to it. The figure below demonstrates how a width shift range works on an input image. The value for the width shift range was slightly higher , just for the demonstration purposes. For this thesis a less value used for this technique, so the important information of the car should not be disturbed.

Figure 3.7: An example of width shift range

From the figure above, it can be seen how width shift range works, but it is highly recommendable to use width shift range with the slightly less values so the rich information of particular image will remain undisturbed.

3.2.7 Height Shift Range

Similar to Width Shift Range, height shift range is a technique which applies vertical shifts or movements to the input image according to the value given to it. Again, it should be noted that slightly use of values are recommended just so the important learning information remains in the input. The figure below shows how a height shift range technique works on a particular image.

Figure 3.8: An example of height shift range

3.2.8 Zoom Range

Zoom range is a technique which applies random zooming to the given input. In CompCar data set, images are randomly zoomed in and out as depicted below.

Figure 3.9: An example of zoomed range

It can be seen in the figure above how an original input image (left) randomly zoomed in and out.

3.2.9 Horizontal Flip

The Horizontal flip as the name suggests flips the entire input image on the horizontal axis. Horizontal axis can also be known as mirroring of an image, as already described in Chapter 2. An example of horizontal flip can be seen in the figure below.

Figure 3.10: An example of Horizontal Flip

3.2.10 Vertical Flip

The vertical flip, flips the given input in vertical axis as shown in the figure below.

Figure 3.11: An example of Vertical Flip

3.2.11 Brightness Range

The brightness range transformation randomly brighten the images. An example can be seen in the below figure.

Figure 3.12: An example of Brightness Range

A sample code below explains how image augmentation is done by using ImageDataGenerator [13].

```
from keras.preprocessing.image import ImageDataGenerator, img_to_array, load_img

datagen = ImageDataGenerator(
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

img = load_img('data/Car.jpg')  # this is a PIL image

# convert image to numpy array with shape (3, width, height)
img_arr = img_to_array(img)

# convert to numpy array with shape (1, 3, width, height)
img_arr = img_arr.reshape((1,) + img_arr.shape)

# the .flow() command below generates batches of randomly transformed images
# and saves the results to the `data/augmented` directory
i = 0
for batch in datagen.flow(
    img_arr,
    batch_size=1,
    save_to_dir='data/augmented',
    save_prefix='Car_A',
    save_format='jpeg'):
    i += 1
    if i > 20:
        break  # otherwise the generator would loop indefinitely
```

Listing 2: Image Augmentation

For the demonstration purposes as seen in the figures above, ImageDataGenerator function is used with each single argument to get the required results.

Chapter 4

Implementation

4.1 Network Architectures

As discussed in the above chapters, Convolutional Neural Networks are used for the classification of different car manufacturers totalling of 163. We have also discussed in the above chapters, how and why we divide these 163 classes data set into three different subsets.

In this section different Convolutional Neural Network architectures that are used in the completion of this thesis will be discussed. In the implementation of CNN architecture we use AlexNet and Inception as mentioned in the paper [45]. Another new CNN architecture i.e. VGG19 that we found to be better in performance than the other two architectures used in the CompCar research paper [45] will also be discuss in the following sections.

4.1.1 Proposed Network Architectures

For the completion of this thesis, the main goal was to identify if CompCar data set is a correct data set that can be use in the classification of cars. For this purpose, following architectures were used to achieve the particular goal.

- Train a model on AlexNet that is able to classify an image into car makes.
- Train a model on Inception architecture that is able to classify an image into car makes a.k.a car manufacturers.
- Train a model on VGG19 to classify an image into car manufacturers.

In each case the data was sub divided into three Subsets i.e. 1000, 2000 and 4000 samples, as discussed in depth in the above chapters.

4.1.2 AlexNet Architecture

AlexNet architecture as discussed in the above chapter was proposed by Alex Krizhevsky in 2012, that was consider to be one of the best CNN architecture in the ImageNet competition.

The AlexNet architecture was trained to predict car manufacturers on all the three cases. The network consists of 5 convolutional layers, 3 fully connected layers are on the top of the network which consists of 2048 nodes and the third fully connected layer consists 1000 nodes followed by the output layer with nodes that should be equal to number of classes on which the model is trained and with the activation function "Softmax". The network takes an RGB image of size 224x224x3 pixels.

For each convolutional layer an activation function Relu is used, following tha MaxPooling2D layer with batch normalization at the end of each layer to normalize the learned data. At the end while compiling the model "Adam" optimizer is used.

Following are the hyperparameters that are being used in AlexNet architecture for each subset of data set.

	HyperParameter	Value
Model	Input Size	224x224
	Batch Size	32
	Optimizer	Adam
	Epochs	60
Preprocessing	Bounding Box	Yes
	Resize	Yes, 224x224
Augmentation	Normalization	Yes, dividing pixels by 255.0
	Image Rotation	0.2
	Width Shift Range	0.2
	Height Shift Range	0.2
	Horizontal Flip	Yes
	Vertical Flip	Yes

Table 4.1: HyperParameters Of AlexNet

4.1.3 InceptionV3 Architecture

The GoogleNet InceptionV3 architecture is one of complex architecture developed and proposed by google which is heavily engineered by researchers. Inception architecture won the image competition in 2014 and played an important role in the development of CNN classifiers.

A pretrained InceptionV3 architecture was used on all the subsets of data set i.e. 1000, 2000 and 4000 samples data set. For training on this model, the first 5 layers were frozen, and a custom layer is added having 2 fully connected layers of 1024 nodes each with Relu function followed by the output layer having the number of nodes equal to number of classes i.e. as described above Top-23,Top-13 and Top-5. Lastly, softmax activation function is used for the classification.

The hyperparameters settings of this model in all the 3 subsets are given below.

	HyperParameter	Value
Model	Input Size	224x224
	Batch Size	32
	Optimizer	SGD
	Epochs	60
Preprocessing	Bounding Box	Yes
	Resize	Yes, 224x224
Augmentation	Normalization	Yes, dividing pixels by 255.0
	Image Rotation	0.2
	Width Shift Range	0.2
	Height Shift Range	0.2
	Horizontal Flip	Yes
	Vertical Flip	Yes

Table 4.2: Hyperparameters Of InceptionV3

4.1.4 VGG19 Architecture

VGG19 is a convolutional neural network that is trained on more than million images from the imagenet database [10]. VGG19 is a neural network consists of 19 deep layers and the pretrained vgg19 can classify 1000 classes or categories of images, for instance keyboard, mouse, planes, cats etc. VGG19 take 224x224 as an input image size.

Similar as InceptionV3, a pretrained VGG19 architecture was used on all of the subsets of data set explained above. For training purposes, the first 5 layers of pretrained vgg19 network were frozen, and a custom layers are added having 2 fully connected layers of 1024 nodes each with Relu function. Lastly, the output layer is added having the number of nodes equal to number of classes followed by softmax activation function is used for classifcation.

The hyperparameter settings for VGG19 used for all subsets are given below.

	HyperParameter	Value
Model	Input Size	224x224
	Batch Size	32
	Optimizer	SGD
	Epochs	60
Preprocessing	Bounding Box	Yes
	Resize	Yes, 224x224
Augmentation	Normalization	Yes, dividing pixels by 255.0
	Image Rotation	0.2
	Width Shift Range	0.2
	Height Shift Range	0.2
	Horizontal Flip	Yes
	Vertical Flip	Yes

Table 4.3: Hyperparameters Of VGG19

4.2 Technology Stack

This section will discuss about the technologies used in the implementation of this thesis. Since a lot of technologies were used for the data preparation, only few important ones will be explained in the following section.

4.2.1 Tensorflow

Figure 4.1: The Official Logo of Tensorflow

Tensorflow is one of the most famous library in use for machine and deep learning algorithms.Tensorflow is an open source library developed by Google in 2015. Tensorflow uses tensors to run computation . Tensors are known to be a generalized set of matrices and vectors which can be recognized as multidimensional arrays. It is a versatile library which can run easily on many platforms like MAc, Windows, Linux etc. As Machine learning and Deep learning algorithms needed more computational power and efficiency, tensorflow emerges to be the library that can effect the training speed of the machine using GPU alongside CPU.

4.2.2 Keras

Figure 4.2: The Official Logo Of Keras

Keras is a neural network api use for developing machine learning algorithms. Keras is a high-level python library which can give the developers a leverage to run it on top of tensorflow or theano. In this thesis keras library is extensively used for deep learning algorithms for number of reasons such as leverage of easy and fast prototyping, support of both CNN and recurrent networks (RN), also the combination of both networks i.e. CNN and RN is also supported [13]. The steps for building a CNN using keras library are as follows.

- Defining the network using keras.models library.
- Compiling the network after adding custom layers.
- Fitting the network.
- Evaluation and Predictions.

4.2.3 Sklearn

Figure 4.3: The Official Logo OF Sklearn

Sklearn [15] a.k.a Scikit learn is a free open source machine learning python library which is famous because of number of various features it offers such as classification, clustering and regression algorithms. In this thesis Sklearn's train_test_split library is used to split the dataset into validation and training sets.

4.2.4 Jupyter Notebook

Figure 4.4: Official Logo Of Jupyter Notebook

Jupyter Notebook [25] is an open-source web-based application that allows developers to code, program compile and share documents that contains live code. Jupyter Notebook is usually use for data visualization and machine learning algorithms. For this thesis, Jupyter Notebook is used for the compilation of algorithms and codes.

4.2.5 Kaggle Kernels

Figure 4.5: Official Logo Of Kaggle

Kaggle is a web application famous in the circle of researcher, scientists and machine learners. Kaggle is owned by google, which is known as a community of data scientists. Kaggle is a database having numerous amount of different data sets which is used for Machine Learning purposes. Kaggle also gives users an opportunity to use kernels. Kernels are same like Jupyter Notebook which runs as a online server to compile the codes and algorithms faster. Kaggle kernels are connected with kaggle servers which gives the developers a leverage to use powerful processing that are connected with kaggle servers with the cloud. Kaggle offers 9 hours of usage per kernel which consists how high CPU's as well as GPU's.

4.2.6 Python

Figure 4.6: Official Logo Of Python

Python is known to be one of the most famous and powerful language that is widely used in informational technology for instance in AI, automation systems, web applications, cloud based applications etc. Python is famous because of ease of use, the fame of python in the area of machine learning is because python supports lots of high-level libraries such as scikit, scipy, Numpy, Matplotlib etc that gives leverage to the developers to code and develop algorithms in few lines of code.

4.3 Testing

For checking the testing performance, testing datasets have been used as described above in Chapter 3. The models are analyzed with and without image augmentation. The training accuracy, loss, validation accuracy and validation loss was closely monitored after each step of epoch.

For predictions, the index having the highest value is considered as the final output of the model. In some cases, top two classes is considered as the final output who have the highest values.

Chapter 5

Experiments And Evaluation

5.1 Convolutional Neural Networks Training

As discussed in earlier chapters, three types of convolutional neural networks were applied on the CompCar data set to analyze and to get the predictions. The three architectures are AlexNet, InceptionV3, and VGG19. These architectures are one of the famous and known to be complex and powerful architectures till to date.

In the following section we will discuss about the training accuracy recorded by applying all of the architectures on different subsets of data set i.e. already discussed in the Chapter 3, we will also discuss about the loss and confusion matrix. Moreover, a general evaluation will be discuss about the prediction of car manufacturers with different models from different categories or classes i.e. Top-5, Top-13, and Top-23.

5.2 Evaluation

In this particular thesis, the main aim was to compare the results on CompCar data set by applying different CNN architectures. The evaluation of all the three architectures as discussed in above chapters is recorded and then predictions were made to get the ground truth. In the following section of this chapter we will explain about the classification accuracy we gained, the loss of each architecture applied on each subset and confusion matrix by showing the facts and figures.

5.2.1 Accuracy and Loss

When training a learning model, one of the main thing that should be kept in mind is that overfitting of the model as we have already discussed in previous Chapters. Avoiding the overfitting results in how well your model is fitting itself on the training data.

As we have already discuss how to prevent overfitting, but the real question that comes to mind is that **"How to find out whether the model is overfitting or not?"**. For the answer of this important question, data scientists came up with solution known as "Cross Validation". Cross Validation is a method where the data is split into two parts as mentioned above i.e. training data set and validation data set. The training data set is used for training of model meaning teaching the model

about the particular categories or classes. While the validation set is used for the evaluation of the model.

The metrics or accuracy on the training set shows how well your model is progressing in terms of learning, whereas validation set tells how good the model is in predicting or classifying the data it has never seen i.e. quality of the model.

Loss is an important part in AI neural networks. Loss determines the inconsistency between predictions and the given real labels. It is a non-negative value which when decreases, increases the robustness and performance of training model.

In the following section of this chapter, accuracy and loss graph will be shown and discuss that is gained on each subset of the CompCar data set.

Subset-I (Top-23)

Subset-I as discussed above consists of 23 classes having more than 1000 samples. We trained AlexNet, InceptionV3 and VGG19 on the subset-I, the resulting graphs can be seen below

- AlexNet:

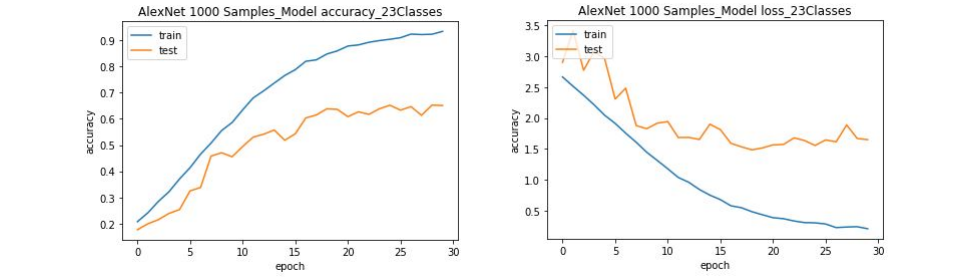


Figure 5.1: Accuracy graph of AlexNet on Top-23

In the figure above, it can be seen that AlexNet architecture on Top-23 classes resulted in training accuracy (acc) of 93% while the validation accuracy (val_acc) is 65%.

- InceptionV3:

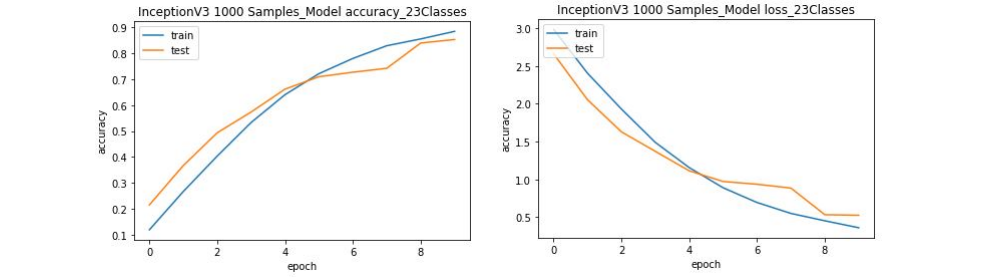


Figure 5.2: Accuracy graph of Inception on Top-23

InceptionV3 on 23 classes resulted in 88% of training accuracy and validation accuracy of 85%.

- VGG19:

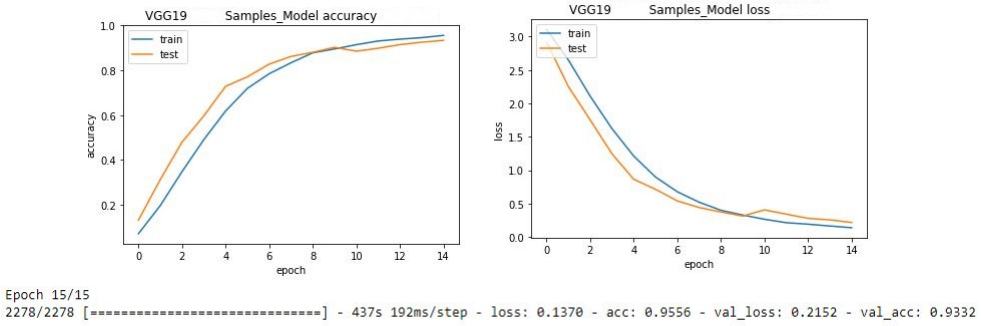


Figure 5.3: Accuracy graph of VGG19 on Top-23

In the figure above it can be seen that VGG19 appeared to be one of the best on Subset-I Top-23 classes as compared to InceptionV3 and AlexNet. The complete accuracy table of each architecture having loss and validation loss of all the architectures can be seen below.

Architecture	loss	Accuracy	Validation loss	Validation Accuracy
VGG19 (Top-23)	0.13	95%	0.21	93%
InceptionV3 (Top-23)	0.35	88%	0.52	85%
AlexNet (Top-23)	0.21	93%	1.6	65%

Table 5.1: Top-23 Accuracy and Loss Table

Architecture	loss	Accuracy	Validation loss	Validation Accuracy
InceptionV3 (Top-13)	0.30	90%	0.38	90%
VGG19 (Top-13)	0.22	92%	0.42	88%

Table 5.2: Top-13 Accuracy and Loss Table

Subset-II (Top-13)

Subset-II consists of Top-13 classes having more than 2000 samples of images each. All of the three architectures were applied on Subset-II also, the graphs of accuracy can be seen below of each architecture.

- InceptionV3:

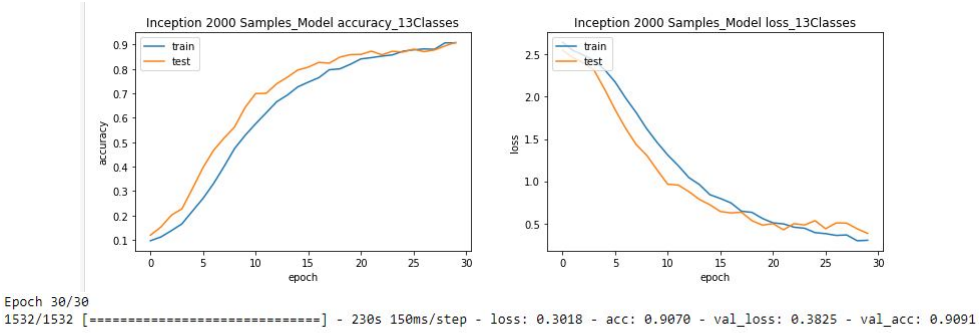


Figure 5.4: Accuracy graph of InceptionV3 on Top-13

On Top-13 InceptionV3 resulted in 90% training accuracy following 90% of validation accuracy.

- VGG19:

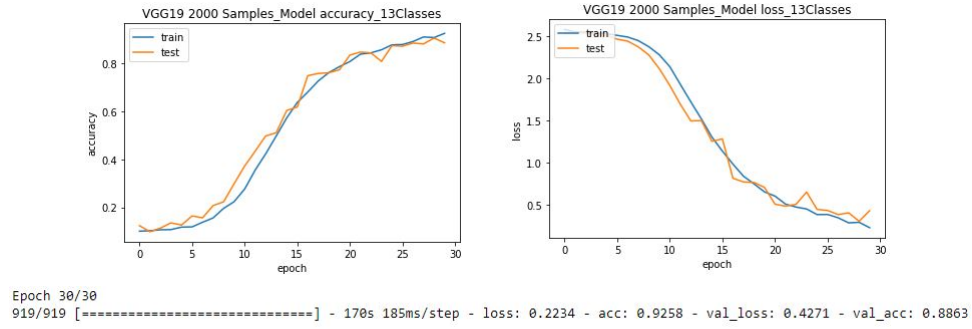


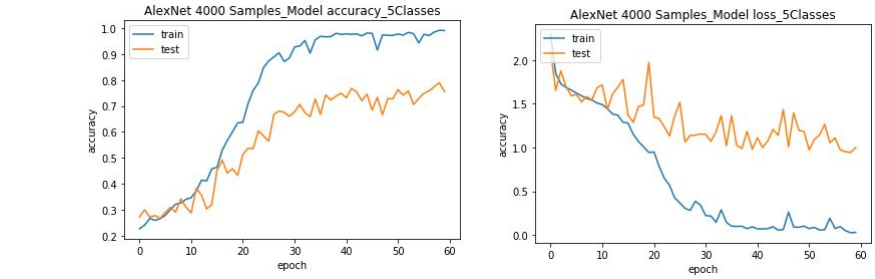
Figure 5.5: Accuracy graph of VGG19 on Top-13

VGG19 on Subset-II Top-13 classes resulted in 92% of training accuracy with validation accuracy of 88%. A complete table having validation loss and training loss can be seen below.

Subset-III (Top-5)

Subset-III consists of Top-5 classes having more than 4000 samples each. Similar to Subset-I and Subset-II, all of the three architectures were applied on this particular subset also, and the results were recorded, that can be seen below.

- AlexNet:

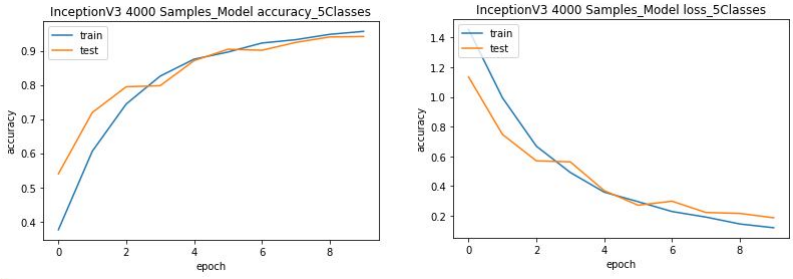



```
Epoch 60/60
4676/4676 [==============================] - 15s 3ms/step - loss: 0.0308 - acc: 0.9908 - val_loss: 1.0007 - val_acc: 0.7556
```

Figure 5.6: Accuracy graph of AlexNet on Top-5

In the figure above, it can be seen that AlexNet architecture on subset-III Top-5 classes having 4000 samples each resulted in a little better performance than the above two subsets, having training accuracy of 99% and validation accuracy of 75%.

- InceptionV3:




```
Epoch 10/10
936/936 [==============================] - 159s 170ms/step - loss: 0.1191 - acc: 0.9558 - val_loss: 0.1848 - val_acc: 0.9416
```

Figure 5.7: Accuracy graph of InceptionV3 on Top-5

With InceptionV3 on Top-5, it was recorded that InceptionV3 resulted with training accuracy of 95% following validation accuracy of 94%.

- VGG19:

Architecture	loss	Accuracy	Validation loss	Validation Accuracy
VGG19 (Top-5)	0.06	98%	0.12	95%
InceptionV3 (Top-5)	0.11	95%	0.18	94%
AlexNet (Top-5)	0.03	99%	1.00	75%

Table 5.3: Top-5 Accuracy and Loss Table

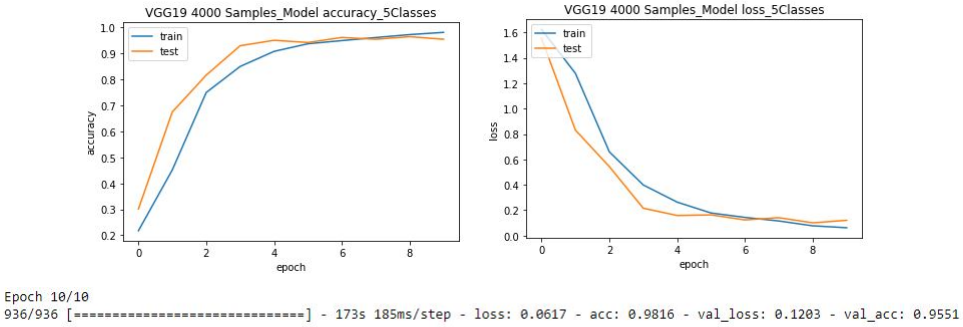


Figure 5.8: Accuracy graph of VGG19 on Top-5

VGG19 again apperead to be one of the best architecture on Top-5 classes taken from Compcar data set. A complete architecture accuracy table having loss and validation loss can be seen below.

5.2.2 Confusion Matrix

Confusion Matrix is a table that evaluate the accuracy of classification problem. It is known to be as a error matrix that visualize the performance of an algorithm. Due to less computing resources below is the only heat map and array of confusion matrix gained from VGG19 on Top-5 classes.

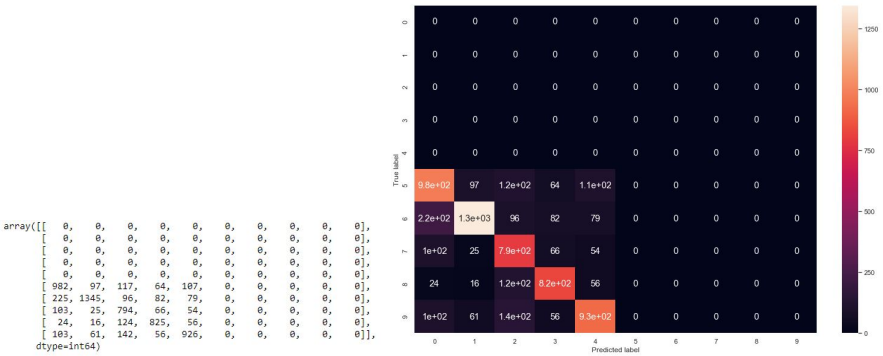


Figure 5.9: Confusion Matrix Heat Map and Array

5.3 General Evaluation

5.3.1 Evaluation of predicting car makes

In this section we will evaluate all the predictions done by the architectures used mainly Inception and VGG19. For the prediction purposes AlexNet is not used, because of low accuracy on all the data set.

5.3.2 Evaluation with VGG19

VGG19 is used to predict number of classes. In this section the evaluation and prediction of VGG19 will be discussed. Because of the time constraint and limited computing resources, only some of the predicted images will be mentioned for each case with the predicted label rather than all the prediction of testing data. As prediction of all the test data requires great amount of computation that is not possible for the available resources for this thesis.

Subset-I (Top-23)

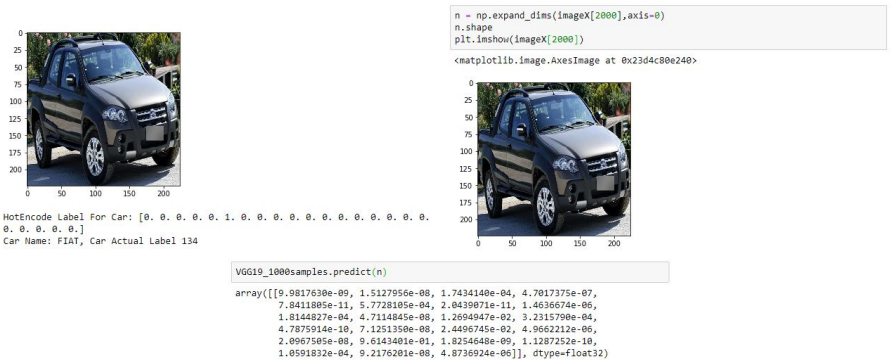


Figure 5.10: Wrong Prediction From VGG19

In the figure above it can be seen that, the car predicted by VGG19 is wrong, whereas the same car predicted by InceptionV3 is right.

Subset-III (Top-5)

Figure 5.11: Right Prediction From VGG19

Figure 5.12: Prediction From VGG19

From the above figures, it can be seen that Mercedes and Volkswagen is given for the prediction in VGG19 architecture. The predicted array shows the number in decimal figures, which as already discussed above that the highest number will be taken as the final prediction, it can be seen that both of the cars from testing data in Top-5 classes is rightly predicted.

5.3.3 Evaluation with InceptionV3

InceptionV3 is also used on the same number of classes as VGG19 to compare the results provided by both the architecture. A brief discussion of predictions obtained from InceptionV3 can be seen below.

Subset-I (Top-23)

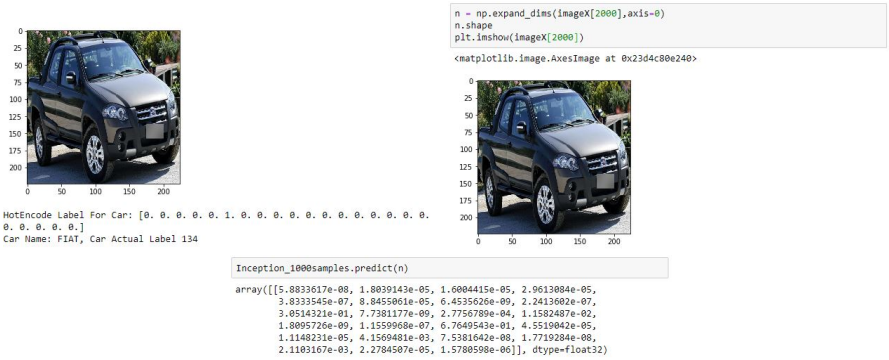


Figure 5.13: Right Prediction From InceptionV3

Figure 5.14: Right Prediction From InceptionV3

Subset-III (Top-5)

Figure 5.15: Right Prediction From InceptionV3

Figure 5.16: Right Prediction From InceptionV3

From the above figures, it can be seen that Mercedes and Volkswagen is given for the prediction in InceptionV3 architecture. The predicted array shows the number in decimal figures, which as already discussed above that the highest number will be taken as the final prediction, it can be seen that both of the cars from testing data in Top-5 classes is rightly predicted.

Chapter 6

Conclusion And Future Work

6.1 Conclusion

This thesis was mainly focused on the classification of images mainly cars using different convolutional neural networks. The thesis was focused on the huge data set provided by CompCar [45] having number of car manufacturers i.e. 163 following 2004 car models. After extensive research it was concluded that Convolutional Neural Networks are the best fit and most powerful networks that can help in visual imagery system to efficiently recognize and classify images. Also, the main and important point in this thesis is to achieve the goal of right predictions with CNNs. The Compcar data set was subdivided into three different subsets for the training of models i.e. number of classes that have more than 1000 image samples each totalling 23 classes, number of classes that have more than 2000 samples each totalling 13 classes, and lastly number of classes that have more than 4000 samples each totalling 5 classes.

This report found a new architecture i.e. VGG19 that have more robust and better results as compared to other two architectures i.e. AlexNet [14] and GoogleNet Inception [34] that have been used in the research paper of CompCar [45].

Furthermore, many image processing techniques were also presented where images were cropped, normalized, shifted and zoomed were discussed in depth. It was observed that without preprocessing of the images the applied CNN does not perform better, therefore the resulting predictions are not really good.

The report also highlights the usage and conversion of a huge data set into readable form and how the bounding boxes were used to get the most important information from an image. Also, the hyperparameters of each model were discussed in depth and explored.

It was also found that VGG19 architecture gave the best results on all the subsets used in this thesis.

It can be concluded that more recent and powerful CNNs can perform more better on other data sets or same having more classes than the classes used in this thesis.

6.2 Future Work

The thesis focused on mainly car manufacturers in the CompCar dataset that was achieved, but this can be further enhanced to Multi-label classification i.e. predicting on Car Makes and car models together with the released year of particular car. As the data set is huge for the use in classifying Car Makes but for using the same data set for Multi-label classification is not enough and insufficient. An interesting and huge data set that came across while researching is Google Open Image Dataset, that have more than 8.5 million images of different cars.

Moreover, another improvement can be done while extending this thesis is the training of a model that has the ability to detect several cars in an image regardless of other objects. This can help overcome the problem of centering of each image that should depict only one car.

Bibliography

[1] History of neural networks, jan.

[2] BBC. The history of machine learning.

[3] Siddhart Das. Convnets architectures:.

[4] Nikhil Deshmukh. Enhancement of communication model for driving simulators by relevant physical effects of radio propagation.

[5] Ben Dickson. What is the aiwinter, nov 2018.

[6] John Burke Ed Burns. artificial neural network (ann), jul 2018.

[7] Alex gray. These charts will change how you see the rise of artificial intelligence, dec 2017.

[8] Data Hacker.

[9] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[10] ImageNet. Vgg19.

[11] Indeed.com. Jobs of the future: Emerging trends in artificial intelligence, aug 2018.

[12] Alexx Kay. artificial neural network (ann), feb 2001.

[13] keras. Image preprocessing.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Scikit Learn. Scikitlearn.

[16] Fu-Jie Huang LeCun, Yann, Y-Lan Boureau, and MarcAurelio Ranzato. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proc. Computer Vision and Pattern Recognition Conference (CVPR07). IEEE Press*, volume 127, 2007.

[17] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95–103, 2011.

[18] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In *European conference on computer vision*, pages 172–185. Springer, 2012.

[19] Fabrízia Medeiros de S Matos and Renata Maria Cardoso R de Souza. Hierarchical classification of vehicle images using nn with conditional adaptive distance. In *International Conference on Neural Information Processing*, pages 745–752. Springer, 2013.

[20] John McCarthy. From here to human-level ai. *Artificial Intelligence*, 171(18):1174–1182, 2007.

[21] Medium. Inceptionv3.

[22] Justin Metz. Why deep learning is suddenly changing your life, sep 2016.

[23] Sunnita Nayak. Understanding alexnet.

[24] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[25] Jupyter Notebook. Jupyter.

[26] quantinsti. Machine learning basics, October 2018.

[27] Margaret Rouse. Artificial intelligence, nov 2010.

[28] Venture Scanner. Artificial intelligence funding trendsq3 2017, sep 2017.

[29] Badreesh Shetty. Supervised machine learning: Classification.

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

[31] Utkrash Sinha. All about biological neuron.

[32] McGuire B. Huang T. & Yang G. Smith, C. History of artificial intelligent [scholarly project], dec 2006.

[33] sushscience. Understanding alexnet.

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[35] Techopedia. Deep neural network.

[36] ujjwalkarn. Convnets explanation.

[37] Amrit Virdee. Data augmentation experimentation.

[38] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[39] Skymind AI Wiki. A beginner's guide to neural networks and deep learning.

[40] Wikipedia. History of cnn.

[41] Wikipedia. Machine learning.

[42] Wikipedia. Supervised and semisupervised, reinforcement.

[43] Wikipedia. Training, validation, and test sets.

[44] Wikipedia. Perceptron, June 2017.

[45] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3973–3981, 2015.

[46] Wei Zhang and Kunio Doi. Shift-invariant artificial neural network for computerized detection of clustered microcalcifications in mammography, March 31 1998. US Patent 5,732,697.

[47] Dongbin Zhao, Yaran Chen, and Le Lv. Deep reinforcement learning with visual attention for vehicle classification. *IEEE Transactions on Cognitive and Developmental Systems*, 9(4):356–367, 2017.

[48] Yiren Zhou and Ngai-Man Cheung. Vehicle classification using transferable deep neural network features. *arXiv preprint arXiv:1601.01145*, 2016.