IMAGE CLASSIFICATION USING DEEP LEARNING

A Project report submitted in partial fulfilment for the award of

Bachelor of Technology

in

Computer Science & Information Technology

Submitted By

Sahdev Saini

(16CS41)

Under Supervision of

Dr. Brajesh Kumar

Department of Computer Science and Information Technology

Faculty of Engineering & Technology M.J.P. Rohilkhand University

Bareilly

2020

UNDERTAKING

We declare that the work presented in this project titled "***Image Classification with deep learning***", submitted to the **Computer Science and Information Technology, Faculty of Engineering & Technology, M.J.P. Rohilkhand University**, Bareilly for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Information Technology***, is my original work. I have not submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

Student Name

Sahdev Saini(16CS41)

DATE-

PLACE-

CERTIFICATE

Certified that the work contained in the project titled "***Image Classification using deep learning***", by Student Name, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Dr.Brajesh Kumar
Computer Science & Information Technology
I.E.T, MJP Rohilkhand University, Bareilly

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher **Dr. Brajesh Kumar & Mr.Triveni pal** who gave me the golden opportunity to do this wonderful project on the topic Image Searching, which also helped me in doing a lot of Research and i came to know about so many newthings I am really thankful to them.Secondly i would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Student Name

Sahdev Saini(16CS41)

REPORT

This is to certified that **Sahdev Saini(16CS41)** are doing project entitled "Image Classification using deep learning" in the partial fulfilment of the requirement for the award of Degree of Bachelor of Technology, under my supervision.

DATE- **Signature**

Dr. Brajesh Kumar

Computer Science & Information Technology

MJP Rohilkhand University (U.P.).

ABSTRACT

Image Classification is a process of identifying a specific object in an image or video sequence. This task is still a challenge for computer vision systems. Many different approaches of object recognition including the traditional classifier or deep neural network were proposed. The objective of this thesis is to implement a deep convolution neural network for object classification. Different architecture and different parameters have been tested in order to improve the classification accuracy. This thesis propose a very simple deep learning network for object classification which comprises only the basic data processing. In the proposed architecture, deep convolution neural network has a total of five hidden layers. After every convolution, there is a subsampling layer which consists of a 2×2 kernel to do average pooling. This can help to reduce the training time and compute complexity of the network. For comparison and better understanding, this work also showed how to fine tune the hyper-parameters of the network in order to obtain a higher degree of classification accuracy. This work achieved a good performance on kaggle dogs vs cats dataset where the accuracy is 76.19%. In challenging image databases such as Pascal and ImageNet, this network might not be sufficient to handle the variability. However, deep convolution neural network can be a valuable baseline for studying advanced deep learning architectures for large-scale image classification tasks. This network can be further improved by adding some validation data and dropout to prevent overfitting.

CONTENT

1.INTRODUCTION

Image Classification is a process of finding and identifying a specific object in a digital image or video sequence. Humans can easily recognize an object in an image even through the object inside the image may vary somewhat in different sizes or scales, different vantage points and even partially obstructed from view. However, object recognition from an image or video is still a challenge for computer vision systems. Even with the help of smart algorithms and human assistants, a classifier in the computer is still unable to catch everything in an image (Sivic and Zisserman, 2003). Many approaches to the task have been implemented over multiple decade. Image Classification task is successful if the network system is able to label the object based on models of known objects. For example, given an image containing one or more different objects with background, the network system is capable of assigning the labels to a set of regions in the image correctly as showed in Figure 1.1. The classification accuracy of the network system can be calculated by comparing the result with a set of labels corresponding to a set of objects known to the system. The object recognition has a very close relationship with segmentation. This is because if the network system is unable to recognize an object, segmentation cannot be done correctly, and without a good segmentation, image classification cannot be done well.

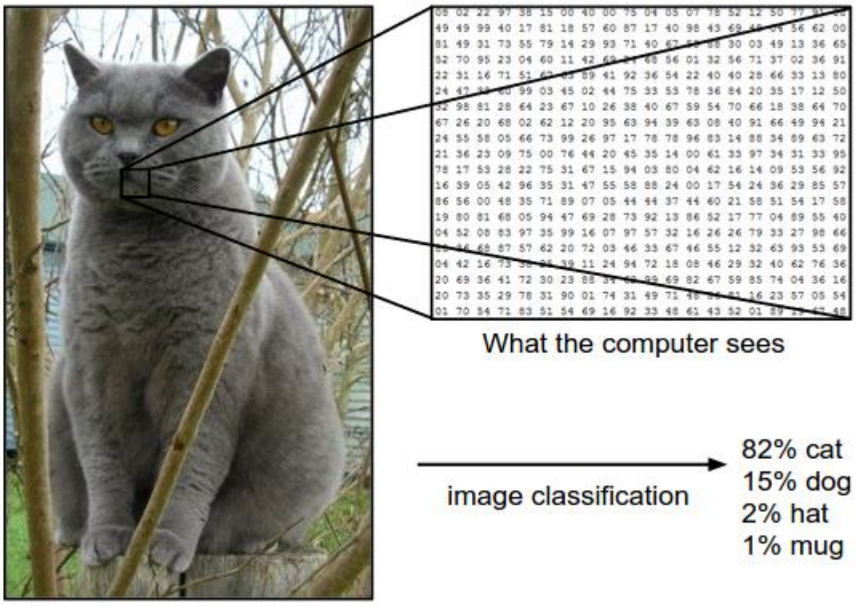


Fig 1.1 Image Classification of cat

Machine learning is a set of algorithms that can learn and explore from the construction and recognize the patterns or objects from an input data. Therefore, machine learning can make accurate predictions for previously unseen data. Hence, machine learning can be used as a powerful tool to overcome the challenges in computer vision such as object recognition, natural language understanding, medical imaging, and web search/information retrieval. In the past few decades, machine learning shows that it can be used in many real-world applications and is successful in solving many artificial intelligence (AI) problems (Lee, 2010). For example, it has been successfully applied in practical speech recognition, effective web search, and face detection.

Machine learning gives a handful of labeled examples and able to do binary classification. For example, given ten images, five images of table with the label zero and another five images of not table with the label one. The algorithm of the system starts to learn and identify images of table. After the training process is done and when new images are fed to the network, the network is able to produces the correct label. In other words, the network produce output zero if the image contains a table, and output one if the image does not contain a table. Recently, deep architectures show a good way to do binary representations by extracting the important features and characterizing of the input distribution. Deep learning also known as deep machine learning, deep structural learning or hierarchical learning is extension algorithms of machine learning that attempts to model higher level of abstractions in data by using complex architectures. The deep learning structural composed of multiple layers and multiple non-linear transformations is used for hierarchical feature (Schmidhuber, 2014). The neural network is shallow if the number of layers of units, regardless of their types, is usually at most two. A deep neural network is deep if it has multiple, usually more than three layers of units. In essence, a neural network is deep when the following two conditions are met. The first condition is the network can be extended by adding layers consisting of multiple units and second condition is the parameters of each layer are trainable (Bengio and LeCun, 2007). From these conditions, it should be understood that there is no absolute number of layers that distinguishes deep neural networks from shallow ones. Rather, the depth of a deep neural network grows by a generic procedure of adding and training one or more layers, until it can properly perform a target task with a given dataset. In other words, the data decide how many layers a deep neural network needs (Cho, Raiko, and Ihler, 2011).

Deep learning tries to move in this direction by capturing a good representation of input data by using compositions of non-linear transformations. A good representation can be defined as one that disentangles underlying factors of variation for input data. It turns out that deep learning approaches can find useful abstract representations of data across many domains (Ainsworth, 2006). Facebook is also planning on using deep learning approaches to understand its users. Deep learning has been so impactful in industry that MIT Technology Review named it as a top-10 breakthrough technology of 2013.

1.1 Classification

Classification involves predicting which class an object,image belongs to.some classifier are binary resulting in yes/no decision.others are multi class,able to categories an image or object in one or several categories.classification algorithm are used to solve problems like email spam filtering,document categorization,image classification & handwritten recognization

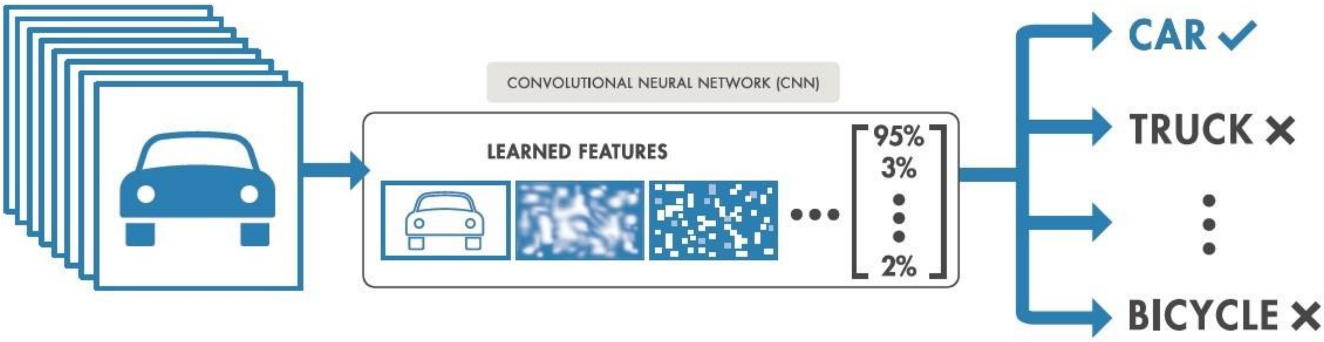


Fig 1.2 Classification of Image

Classification is a systematic arrangement in groups and categories based on its features. Image classification came into existence for decreasing the gap between the computer vision and human vision by training the computer with the data. The image classification is achieved by differentiating the image into the prescribed category based on the content of the vision. In deep learning, we consider the neural networks that identify the image based on its features. This is accomplished for the building of a complete feature extraction model which is capable of solving the difficulties faced due to the conventional methods. The extrac- tor of the integrated model should be able to learn extracting the differentiating features from the training set of images accurately. Many methods like GIST, histogram of gradient oriented and Local Binary Patterns, SIFT are used to classify the feature descriptors from the image.

1.2 Neural Network

A neural network is defined as a computing system that consist of a number of simple but highly interconnected elements or nodes, called 'neurons', which are organized in layers which process information using dynamic state responses to external inputs. This algorithm is extremely useful, as we will explain later, in finding patterns that are too complex for being manually extracted and taught to recognize to the machine. In the context of this structure, patterns are introduced to the neural network by the input layer that has one neuron for each component present in the input data and is communicated to one or more *hidden layer* present in the network; called 'hidden' only due to the fact that they do not constitute the input or output layer. It is in the hidden layers where all the processing actually happens through a system of connections characterized by ***weights and biases*** *(commonly referred as W and b)*: the input is received, the neuron calculate a weighted sum adding also the bias and according to the result and a pre-set **activation function** (most common one is sigmoid, *σ,* even though it almost not used anymore and there are better ones like ReLu*)*, it decides whether it should be 'fired' or activated. Afterwards, the neuron transmit the information downstream to other connected neurons in a process called '*forward pass*'. At the end of this process, the last hidden layer is linked to the *output layer* which has one neuron for each possible desired output.

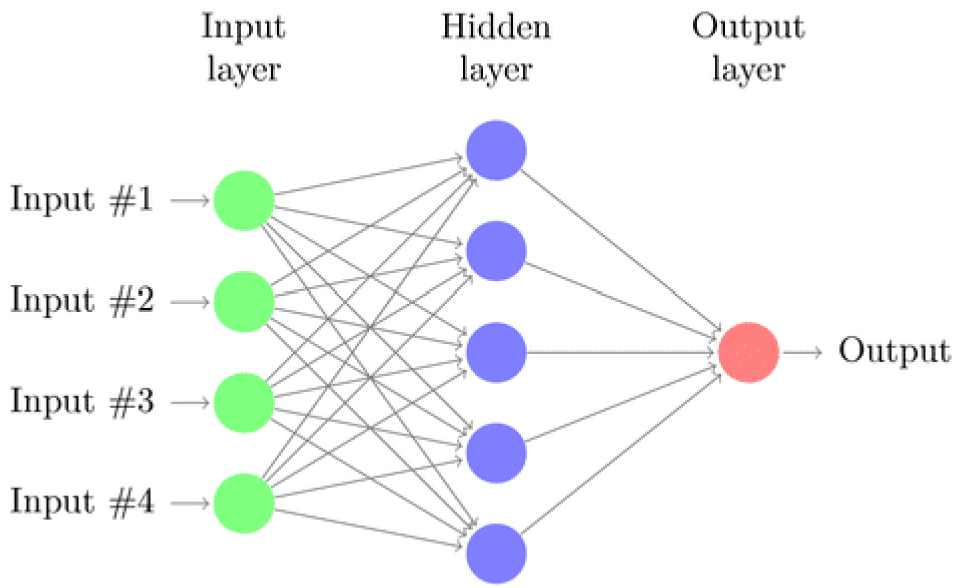


Fig1.3 neural network

1.3 Architecture of neural networks

The design of NNs is inspired by the way the brain works. You shouldn't overstretch this point; it's a loose inspiration. The brain is a network of neurons. The human brain has about one hundred billion neurons, and each neuron is, on average, connected with ten thousand other neurons. Let’s take a look at the brain’s basic unit—a neuron.

Figure 2 shows a simplified sketch of a neuron. It receives signals from other neurons via its dendrites. Some inputs have an activating impact, and some inputs have an inhibiting impact. The received signal is accumulated and processed within the cell body of the neuron. If the signal is strong enough, the neuron fires. That means it produces a signal which is transported to the axon terminals. Each axon terminal connects to another neuron. Some connections can be stronger than others, which makes it easier to transduce the signal to the next neuron. The strength of these connections can be changed by experiences and learning.

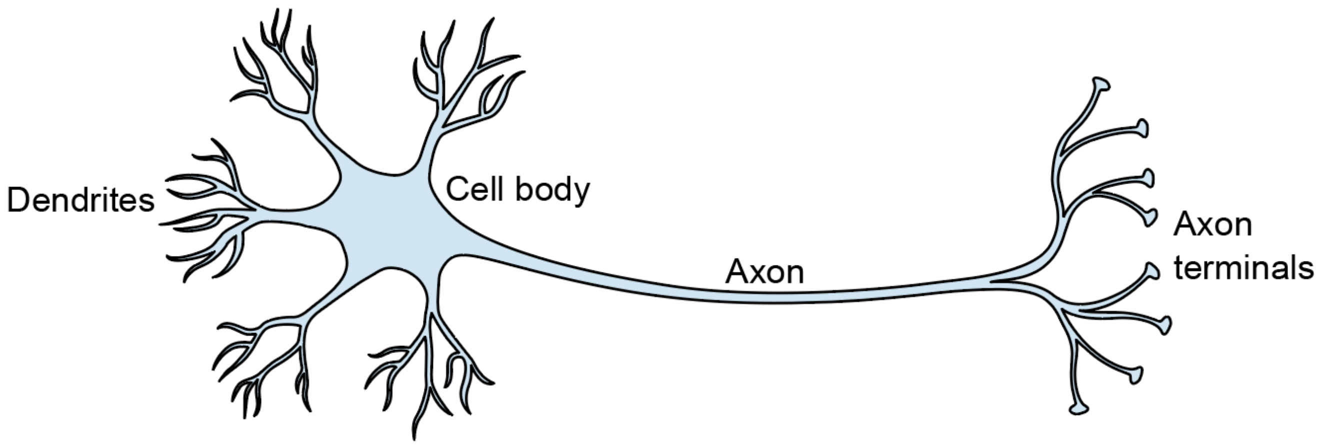
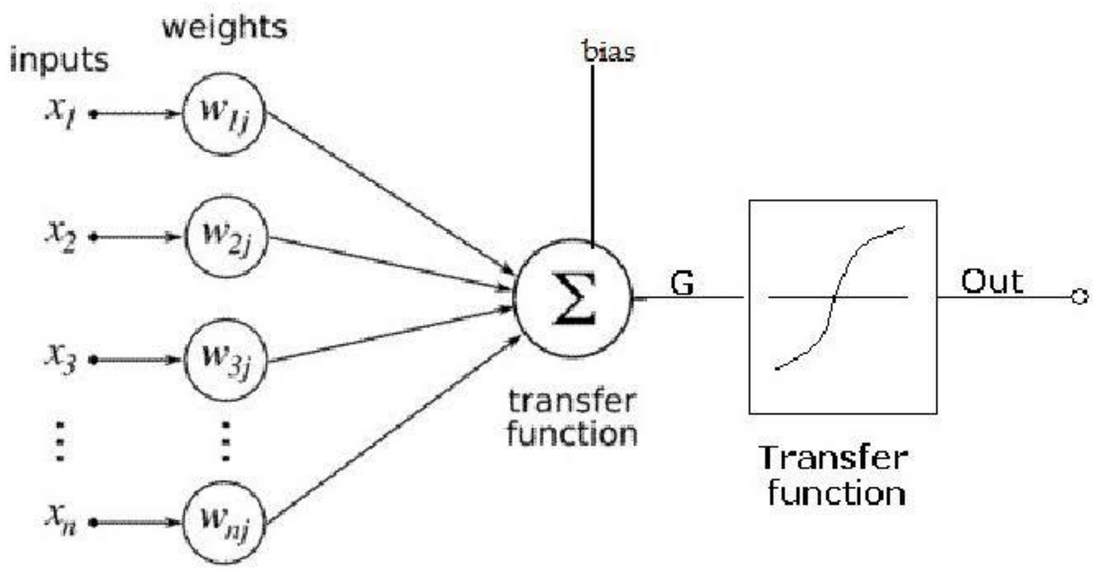


Fig1.4 human brain single biological cell

Computer scientists have derived a mathematical abstraction from the biological brain cell, the artificial neuron shown in figure 3



1.3.1 PERCEPRTON

A perceptron is a neural network unit (an artificial neuron) that does certain computations to detect features or business intelligence in the input data. Perceptron was introduced by Frank Rosenblatt in 1957.

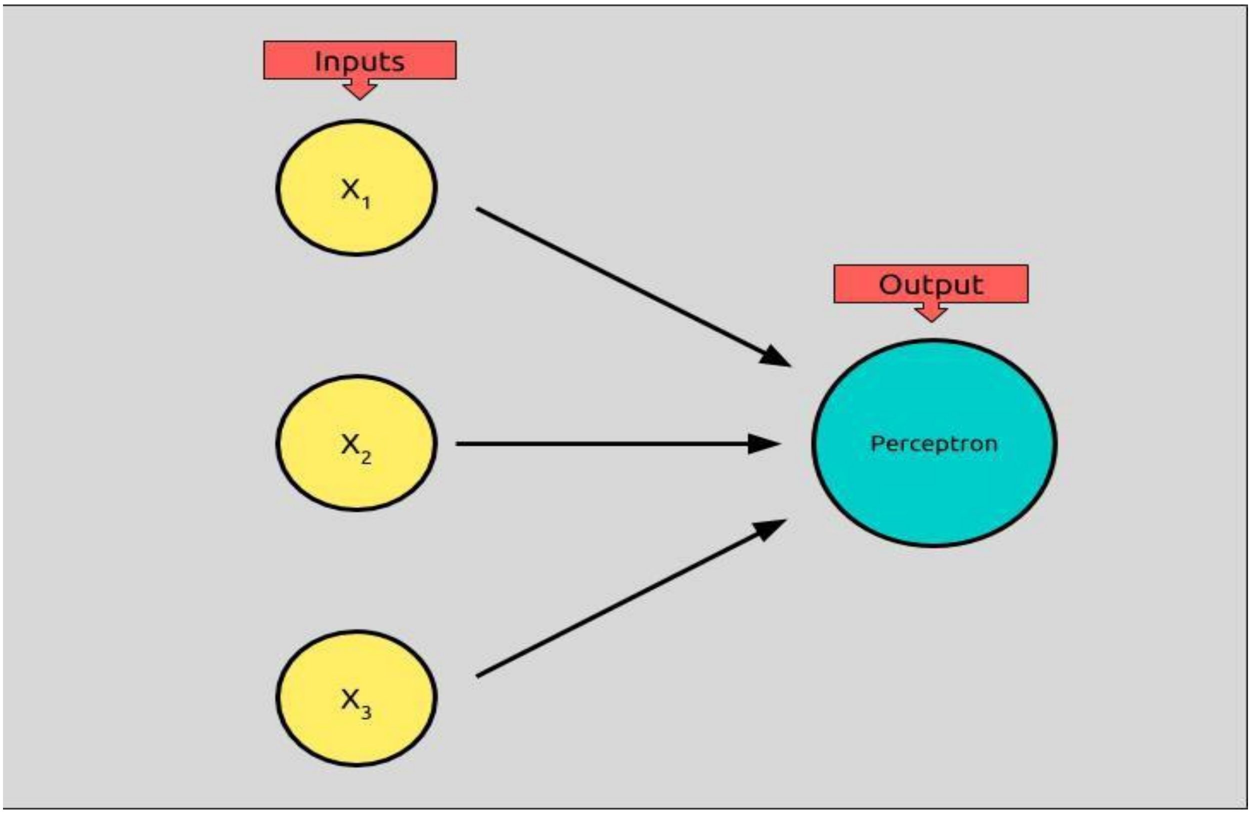


Fig1.5 perceptron model

A perceptron can have any no of inputs ,but this one percpetron above is 3 number of inputs named as x1,x2 to X3.and produces a binary output known as activation function

In the above picture weights are illustrated by black arrows.we will call each weight as w.each input x has associated weights individually.for example x1 has weight w1,x2 has weight w2 &x3 has weight w3.

To determine the perceptron's activation, we take the weighted sum of each of the inputs and then determine if it is above or below a certain threshold, or bias, represented by b.

1.3.1(a) PERCEPTRON LEARNING RULE

Perceptron Learning Rule states that the algorithm would automatically learn the optimal weight coefficients. The input features are then multiplied with these weights to determine if a neuron fires or not.

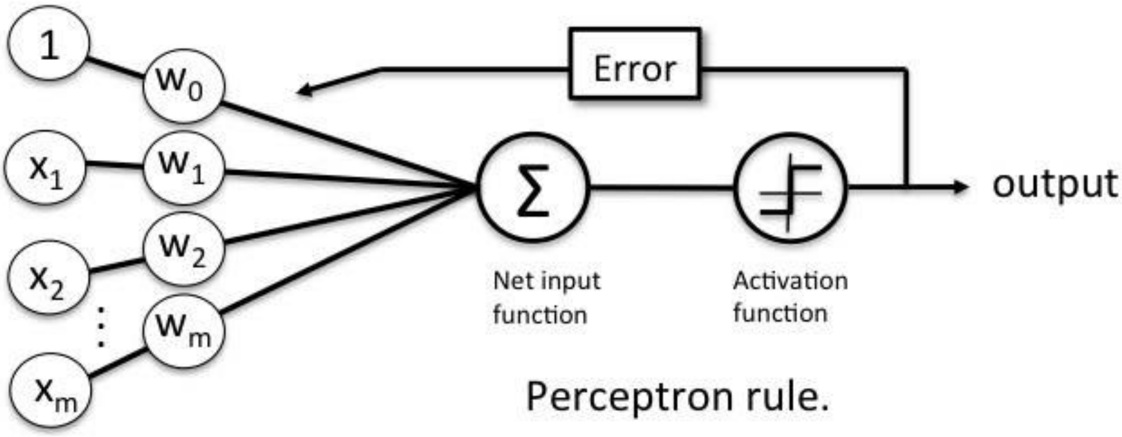


Fig1.6 Perceptron rule

The Perceptron receives multiple input signals, and if the sum of the input signals exceeds a certain threshold, it either outputs a signal or does not return an output. In the context of supervised learning and classification, this can then be used to predict the class of a sample.

1.3.1(b) PERCEPTRON FUNCTION

Perceptron is a function that maps its input “x,” which is multiplied with the learned weight coefficient; an output value ”f(x)”is generated.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

In the equation given above:

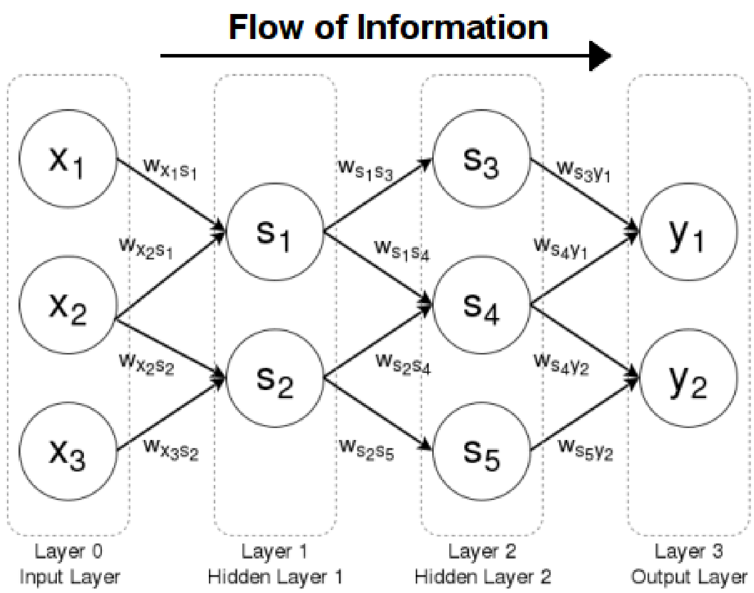
“w” = vector of real-valued weights

“b” = bias (an element that adjusts the boundary away from origin without any dependence on the input value)

“x” = vector of input x values

1.3.2 FEEDFORWARD NEURAL NETWORK

Feedforward neural networks are also known as **Multi-layered Network of Neurons** (MLN). These networks of models are called feedforward because the information only travels forward in the neural network, through the input nodes then through the hidden layers (single or many layers) and finally through the output nodes. In MLN there are no feedback connections such that the output of the network is fed back into itself. These networks are represented by a combination of many simpler models(sigmoid neurons).



S

Fig1.7 : feed forward network

The **main goal** of a feedforward network is to approximate some function f*. For example, a regression function y = f *(x) maps an input x to a value y. A feedforward network defines a mapping y = f (x; θ)

and learns the value of the parameters θ that result in the best function approximation.

The reason these networks are called feedforward is that the flow of information takes place in the forward direction, as x is used to calculate some intermediate function in the hidden layer which in turn is used to calculate y. In this, if we add feedback from the last hidden layer to the first hidden layer it would represent a recurrent neural network.

1.3.3 ACTIVATION FUNCTIONS

Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

We know, neural network has neurons that work in correspondence of *weight*, *bias* and their respective activation function. In a neural network, we would update the weights and biases of the neurons on the basis of the error at the output. This process is known as *back-propagation*. Activation functions make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases. A neural network without an activation function is essentially just a linear regression model. The activation function does the non-linear transformation to the input making it capable to learn and perform more complex tasks.

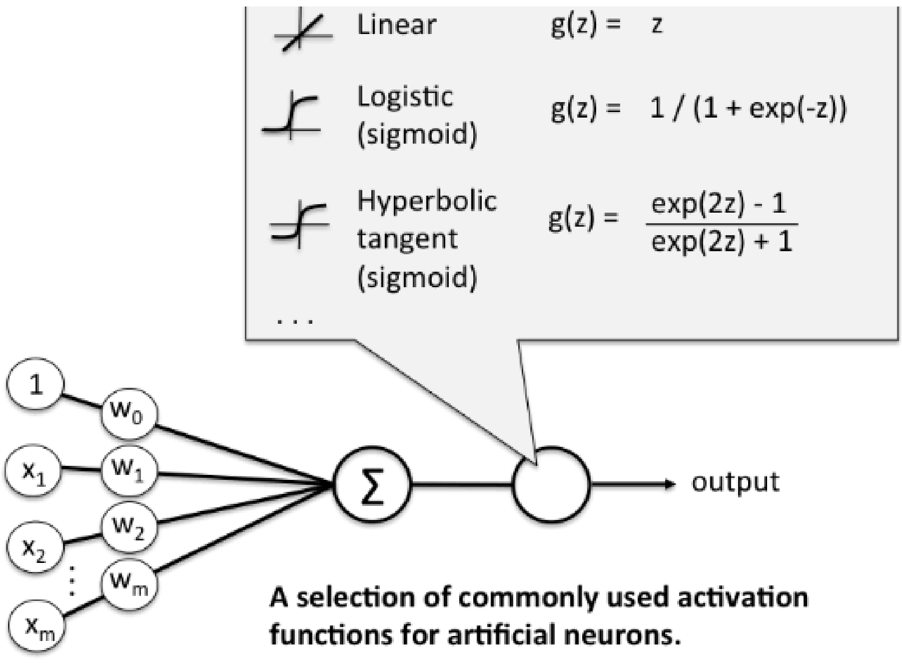


Fig 1.8 activation function

1.3.4 INPUT LAYER

The input layer is responsible for receiving the inputs. These inputs can be loaded from an external source such as a web service or a csv file. There must always be one input layer in a neural network. The input layer takes in the inputs, performs the calculations via its neurons and then the output is transmitted onto the subsequent layers.

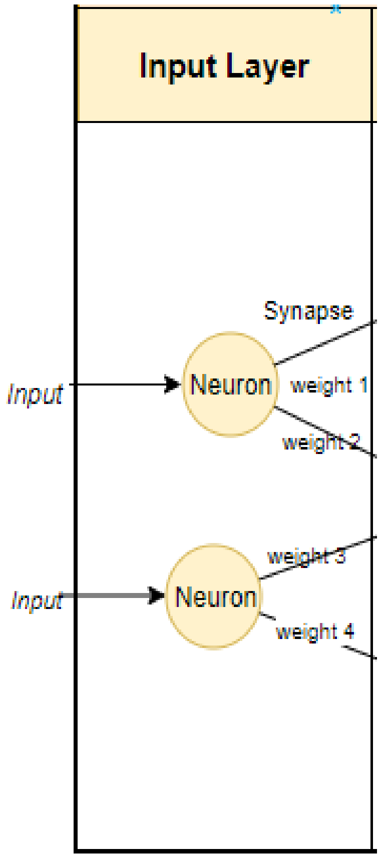


Fig: 1.9 Input layer

1.3.5 OUTPUT LAYER

The output layer is responsible for producing the final result. There must always be one output layer in a neural network.

The output layer takes in the inputs which are passed in from the layers before it, performs the calculations via its neurons and then the output is computed.

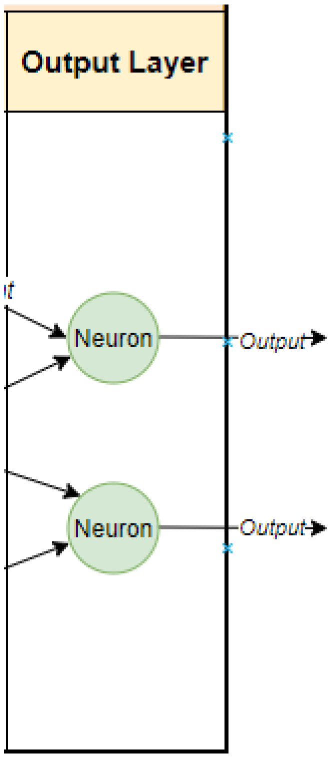


Fig 1.10 output Layer

1.3.6 HIDDEN LAYER

The hidden layer is a layer which is hidden in between input and output layers since the output of one layer is the input of another layer. The hidden layers perform computations on the weighted inputs and produce net input which is then applied with activation functions to produce the actual output. The computations that the hidden layers perform (the way the hidden layers are setup) and the activation functions used depend on the type of neural network used which in turn depends on the application.

The word “hidden” implies that they are not visible to the external systems and are “private” to the neural network. There could be zero or more hidden layers in a neural network. One hidden layer is sufficient for the large majority of problems.usually each hidden layer contaion same number of neurons. The larger the number of hidden layers in a neural network, the longer it will take for the neural network to produce the output and the more complex problems the neural network can solve.

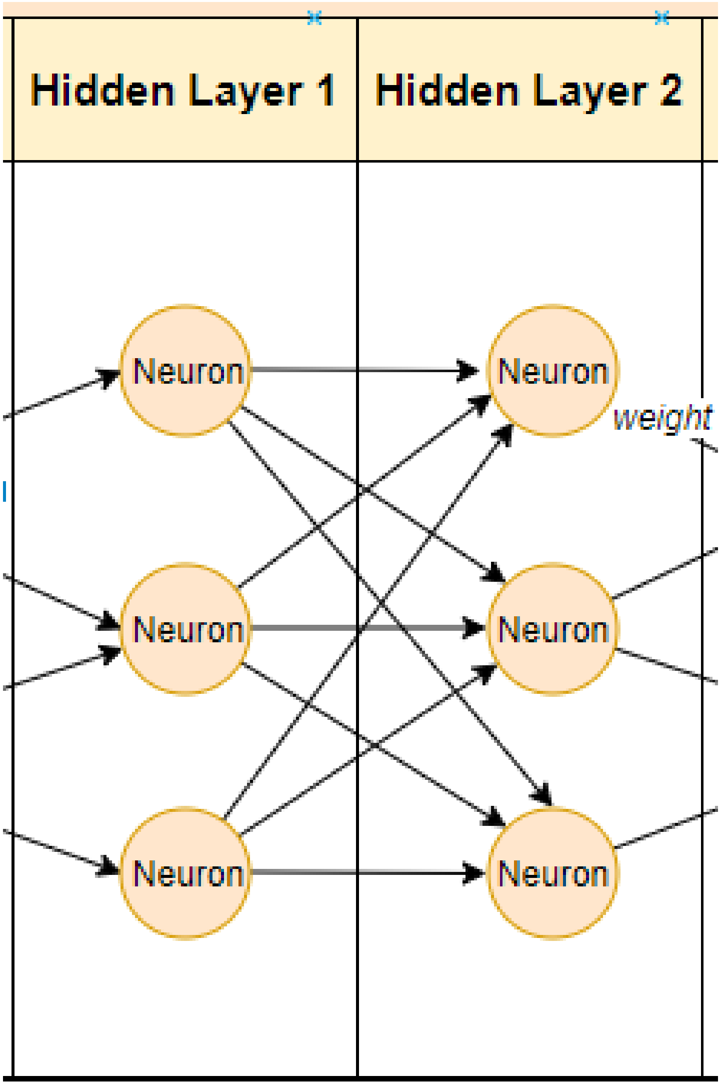


Fig 1.11 Hidden Layer

Hidden layers, simply put, are layers of mathematical functions each designed to produce an output specific to an intended result. For example, some forms of hidden layers are known as squashing functions. These functions are particularly useful when the intended output of the algorithm is a probability because they take an input and produce an output value between 0 and 1, the range for defining probability.

Hidden layers allow for the function of a neural network to be broken down into specific transformations of the data. Each hidden layer function is specialized to produce a defined output. For example, a hidden layer functions that are used to identify human eyes and ears may be used in conjunction by subsequent layers to identify faces in images. While the functions to identify eyes alone are not enough to independently recognize objects, they can function jointly within a neural network.

2. DEEP LEARNING

The field of artificial intelligence is essentially when machines can do tasks that typically require human intelligence. It encompasses machine learning, where machines can learn by experience and acquire skills without human involvement. Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data. Also known as deep neural learning or deep neural network.

2.1 HOW DEEP LEARNING WORKS

Every Deep Learning system works in two phases or steps. First step is training and second is inferring. A useful analogy to understand both these steps will be a graduate student's transition from university to professional work. The training is what happens in a university when the student is exposed to a lot of knowledge. However, while the student is learning a lot, the knowledge is often based on fictional examples rather than a real world problem that needs to be solved. This later more useful thing, i.e. solving a real world problem happens in the second stage, i.e. Inferring. In the training phase the ANNs are exposed to a lot of data. Once the training has taken place, ANNs use inferring to make educated guess on new, previously unexposed, real world data.

Returning to the example of our cat image. In the training phase the ANNs will be exposed to a lot of cat images. They will learn to understand the shape of a cat in different frames and situations. This is also called supervised learning. The ANNs are being shown and told how to identify cats much like you tell your 2 year old kid. In inferring the ANNs will use what they have learnt and make educated guesses if a particular image is a cat or not. Much like when your kid sees a new cat on the street and immediately concludes that it's a cat because you have show her the one at home.

Returning to the example of our cat image. In the training phase the ANNs will be exposed to a lot of cat images. They will learn to understand the shape of a cat in different frames and situations. This is also called supervised learning. The ANNs are being shown and told how to identify cats much like you tell your 2 year old kid. In inferring the ANNs will use what they have learnt and make educated guesses if a particular image is a cat or not. Much like when your kid sees a new cat on the street and immediately concludes that it's a cat because you have show her the one at home. Most modern deep learning models are based on artificial neural networks, specifically, Convolutional Neural Networks (CNN)s, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines. In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level Of course, this does not completely eliminate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.

The word deep in deep learning refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial *credit assignment path* (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited. Deep learning architectures can be constructed with a greedy layer-by-layer method.[Deep learning helps to disentangle these abstractions and pick out which features improve performance.[For supervised learning tasks, deep learning methods eliminate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation. Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data are more abundant than the labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks. A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets. DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them.

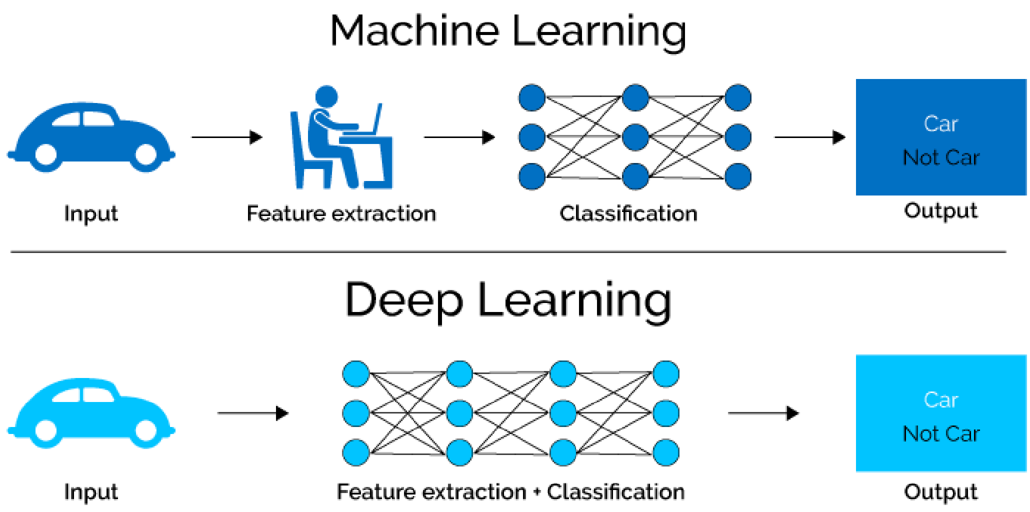


Fig 2 Machine Learning

3. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks (CNN) is one of the variants of neural networks used heavily in the field of Computer Vision. It derives its name from the type of hidden layers it consists of. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers, and normalization layers. Here it simply means that instead of using the normal activation functions defined above, convolution and pooling functions are used as activation functions.

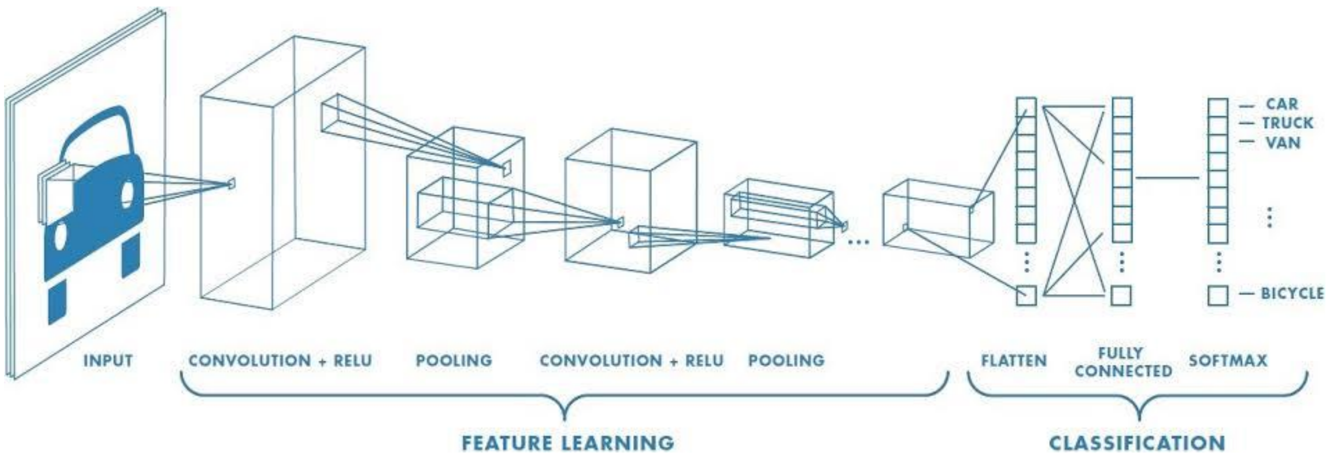


Fig 3 Convolutional Neural Network

3.1 CONVOLUTION

Convolution operates on two signals (in 1D) or two images (in 2D): you can think of one as the "input" signal (or image), and the other (called the kernel) as a "filter" on the input image, producing an output image (so convolution takes two images as input and produces a third as output).

In layman terms it takes in an input signal and applies a filter over it, essentially multiplies the input si

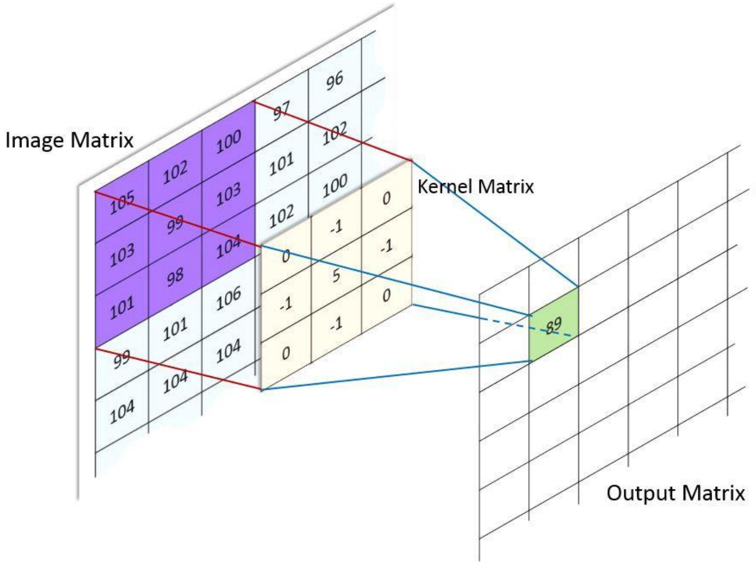


Fig 3.1 Matrix representation

3.2 POOLING

Pooling: Pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.

There are 2 main types of pooling commonly known as max and min pooling. As the name suggests max pooling is based on picking up the maximum value from the selected region and min pooling is based on picking up the minimum value from the selected region.

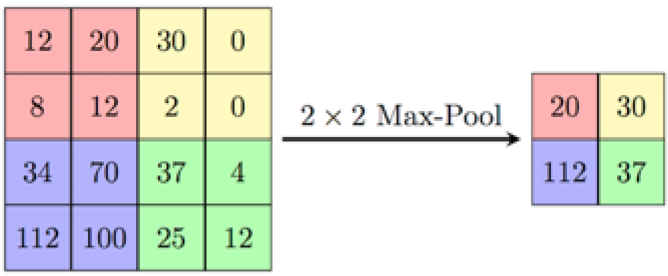


Fig 3.2 Max-Pool

4. METHODOLOGY

The requirements material to carry out this project is as follows.

4.1. Hardware Requirements:

1.Intel based processor

2.RAM

3.HDD

4.2.Software Requirements

1. Window operating system

2. Anaconda Navigator

3. Juypter Notebook

4.3 Library Used

1. Keras Library Of Python

4.3.1 Layers Of Keras Used

1. Conventional

2. Maxpooling

3. Flatten

4. Dense

5. The working procedure of the project is as

1. The images take in folder and create a large dataset.

2. Store the features in the csv file.

3. Take the image in the query folders and execute main file.

4. Than we get the similar images which we given in query image.

6. RELATED WORK

6.1.CREATE

Create a folder datasets

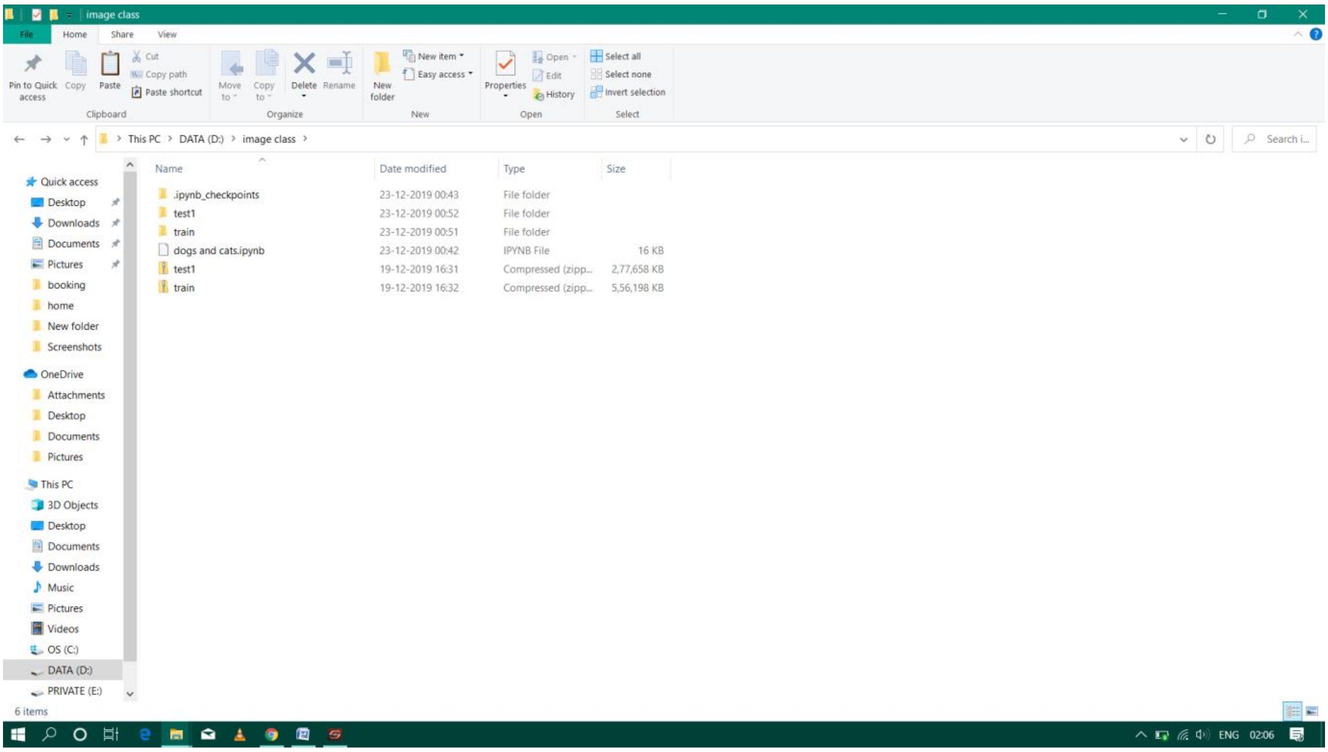


Fig 6 Data Set folder

6.2 DATA SET

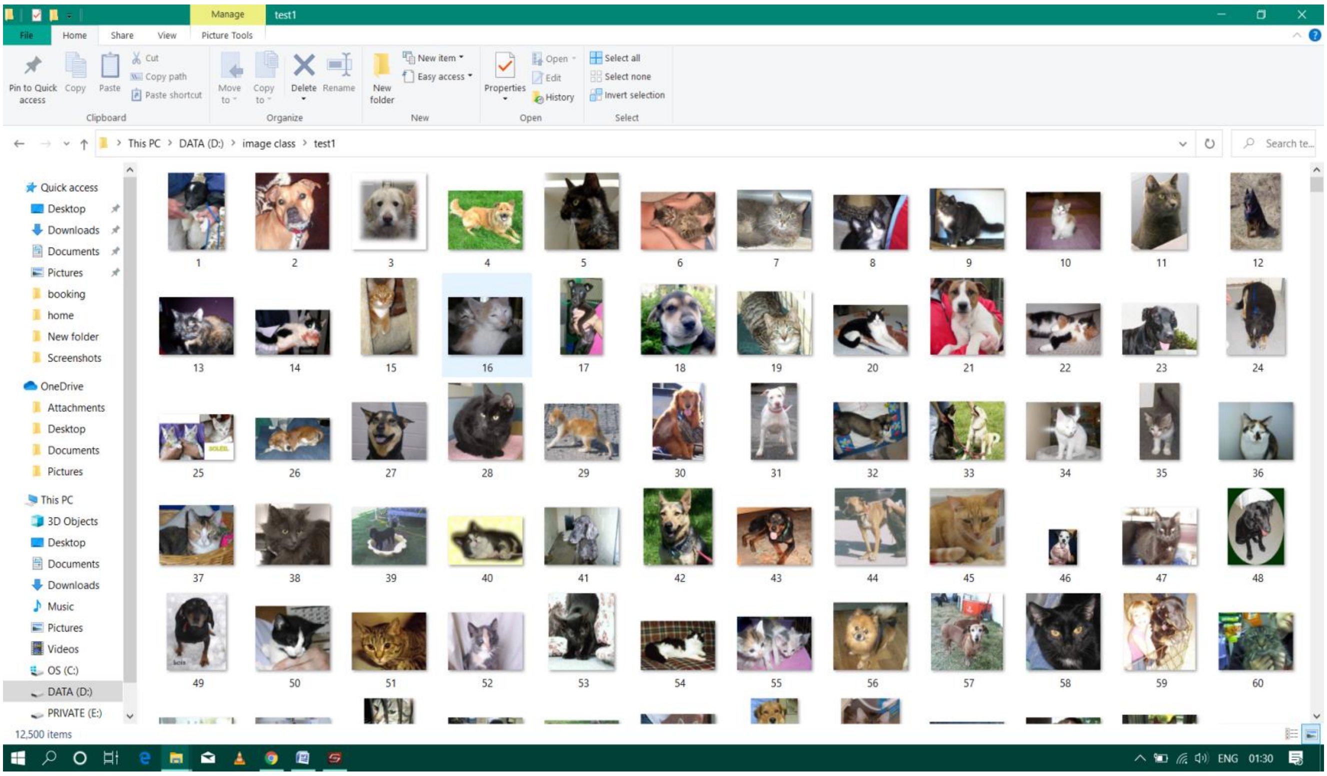


Fig 6.1 Data set

For training our CNN model we uses the kaggle dogs vs cats datatset. This dataset includes the 1000 trained images of cats and 900 trained images of dogs.and we uses 1000 images of dogs & cats for test.

6.3 ANACONDA NAVIGATOR

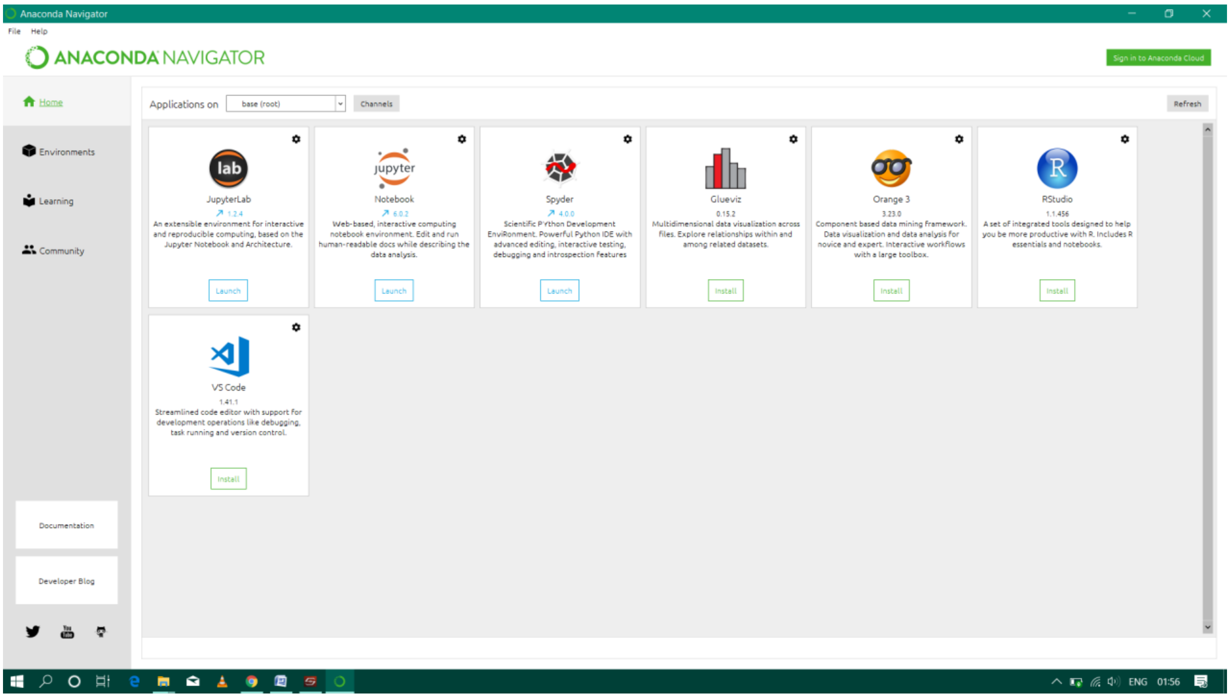


Fig 6.2 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

6.4 IMPORT LIBRARY

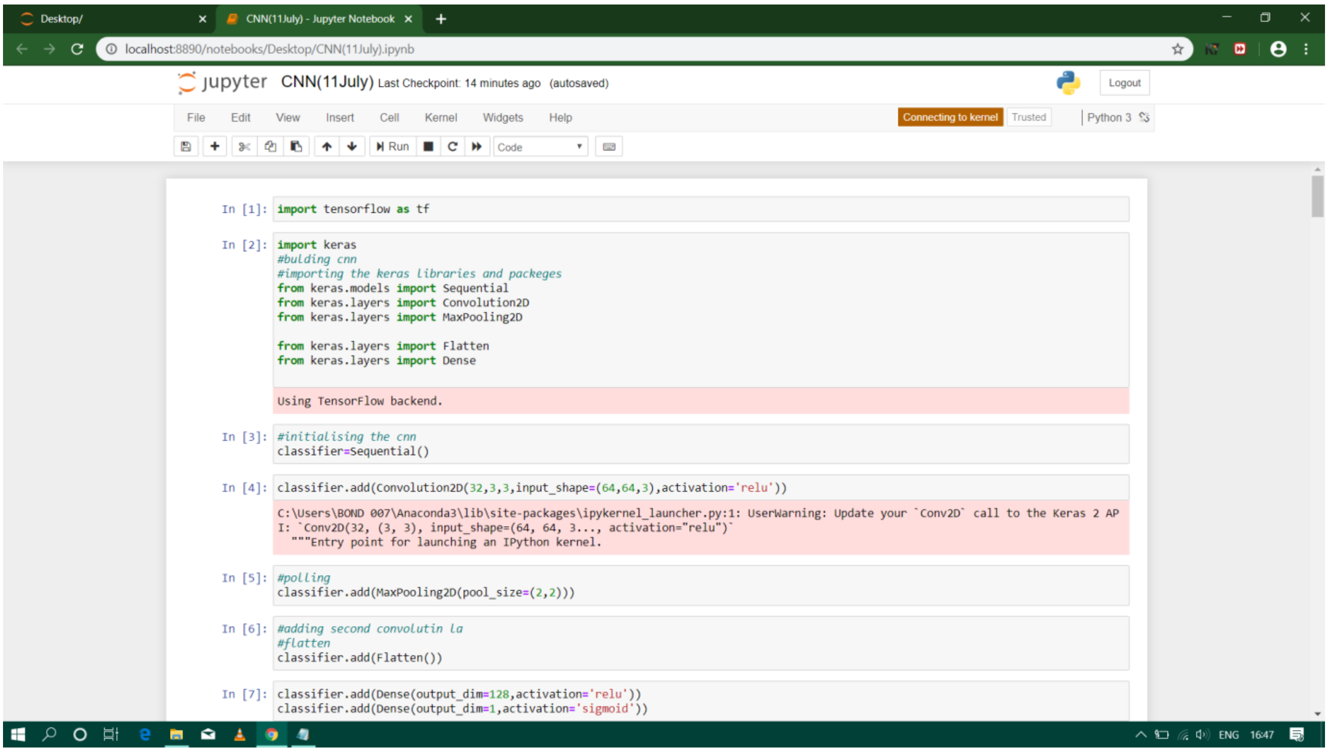


Fig 6.3 Import Library

Keras is a popular and user-friendly deep learning library written in Python. The intuitive API of Keras makes defining and running your deep learning models in Python easy. Keras allows you to choose which lower-level library it runs on, but provides a unified API for each such backend. Currently, Keras supports Tensorflow, CNTK and Theano backends.

Keras model import is targeted at users mainly familiar with writing their models in Python with Keras. With model import you can bring your Python models to production by allowing users to import their models into the DL4J ecosphere for either further training or evaluation purposes

6.5 EPOCH SETS

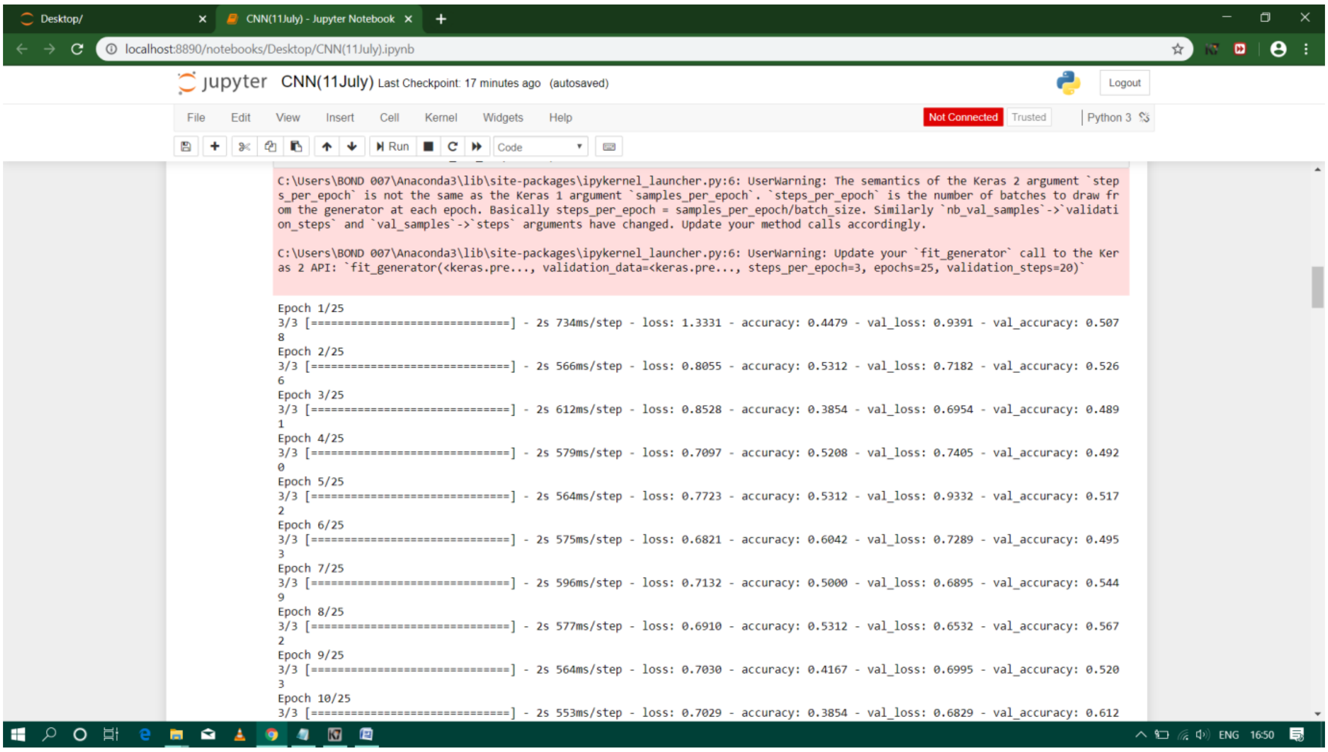
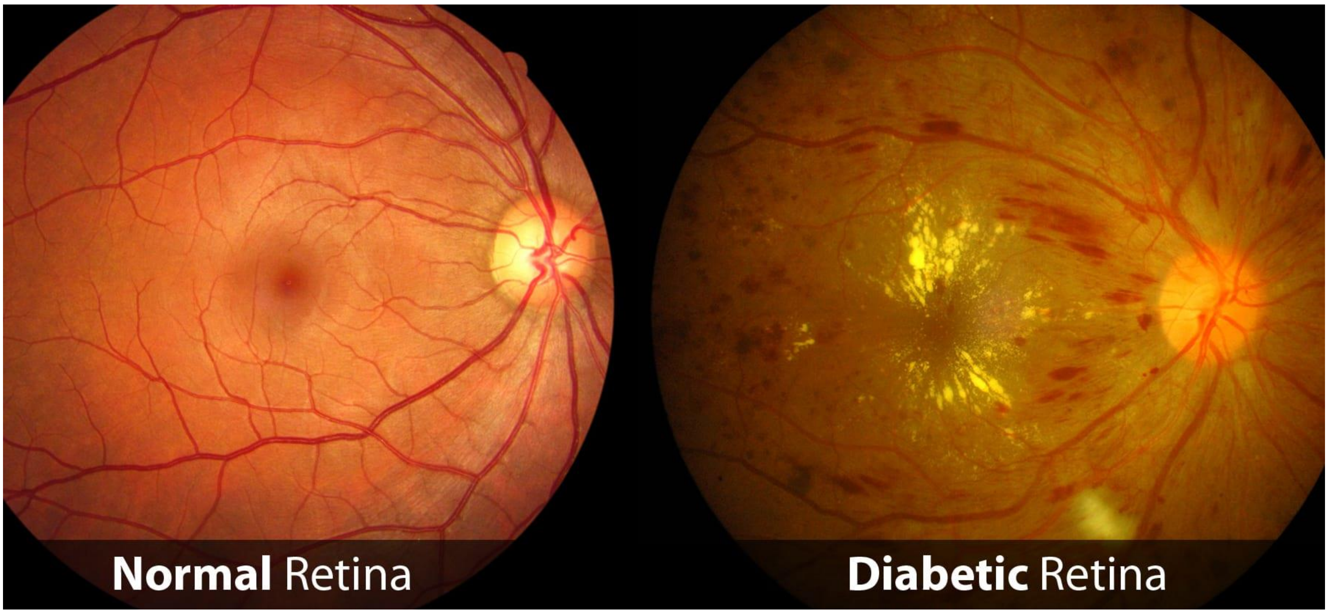


Fig 6.5 Epoch set

7. Applications of Image Classification:

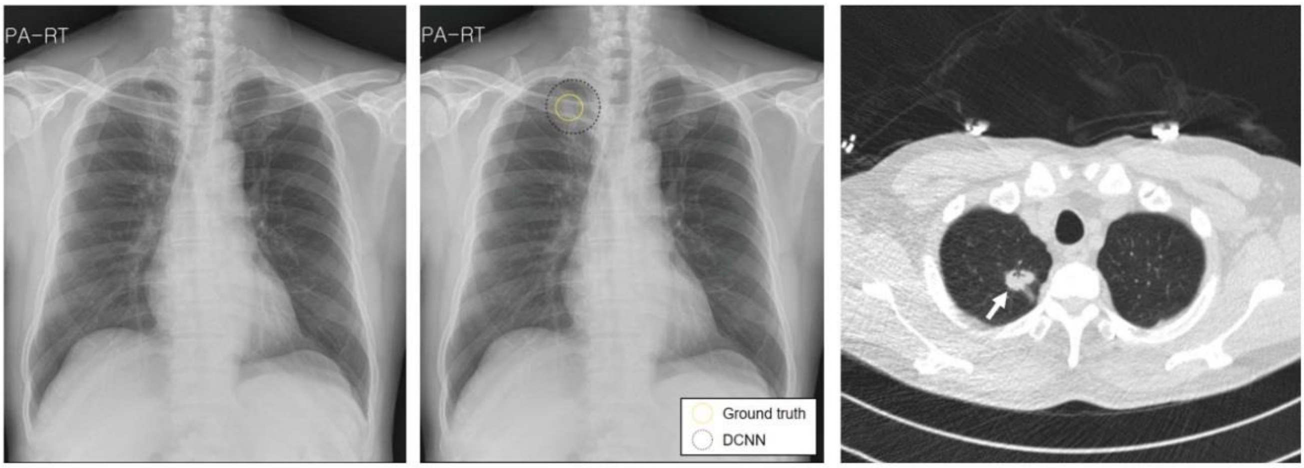
7.1 Detecting diabetic eye disease

The deep learning model detects the severity grade of diabetic retinopathy and macular edema accurately. Diabetic retinopathy is one of the most common comorbidities of diabetes that, if untreated, may lead to severe vision loss. Macular edema refers to swelling under a specific part of the retina caused by diabetic retinopathy.



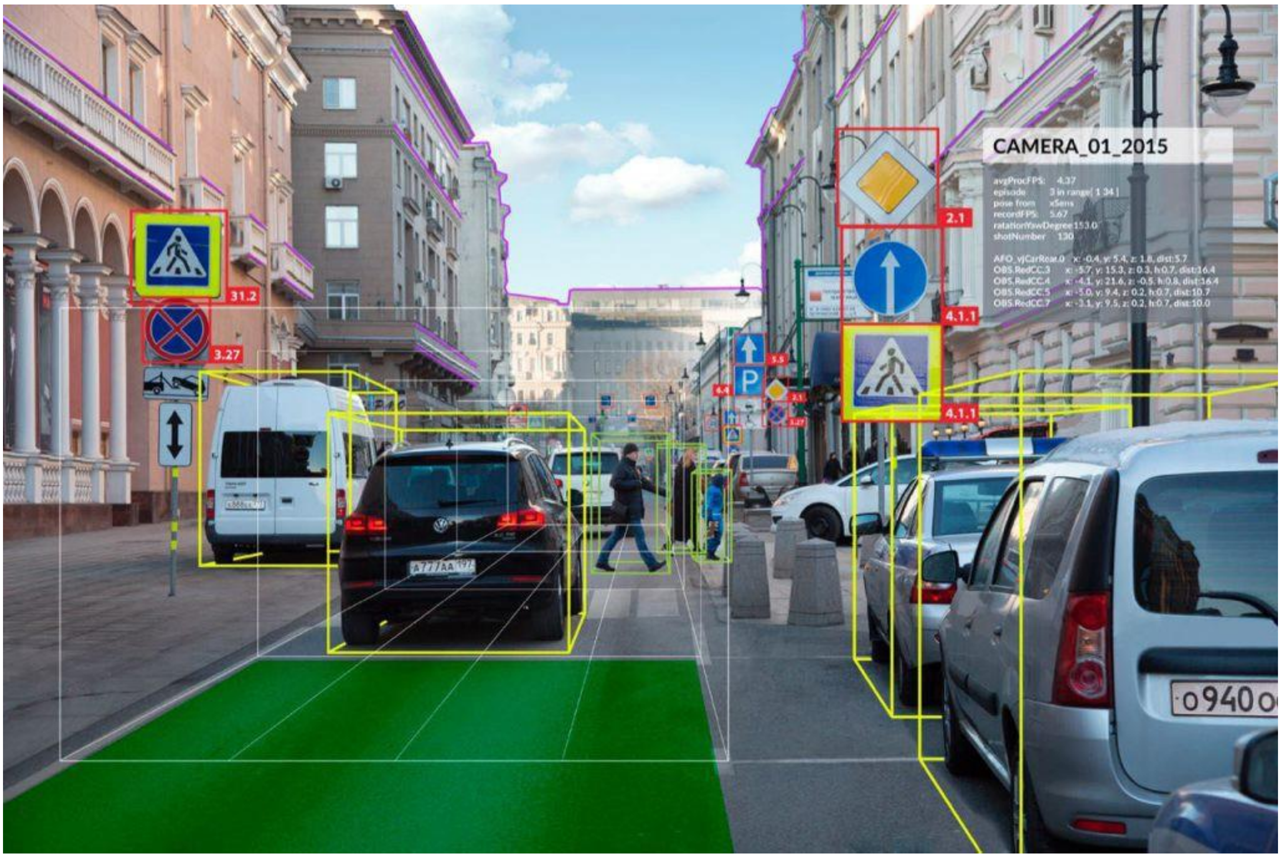
7.2 Lung cancer detection

Lung cancer is the most common cancer that cannot be ignored and cause death with late health care. Currently, CT can be used to help doctors detect the lung cancer in the early stages. In many cases, the diagnosis of identifying the lung cancer depends on the experience of doctors, which may ignore some patients and cause some problems. Deep learning has been proved as a popular and powerful method in many medical imaging diagnosis areas.



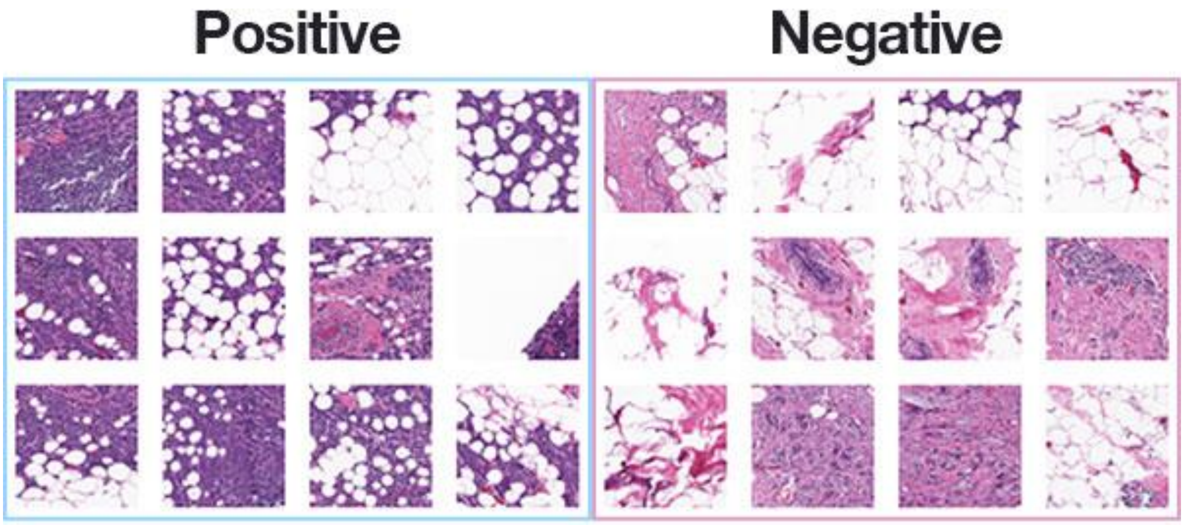
7.3 Self driving cars

Self-driving cars have rapidly become one of the most transformative technologies to emerge. Fuelled by Deep Learning algorithms, they are continuously driving our society forward and creating new opportunities in the mobility sector.image classification is used to built a fully functional self-driving car fuelled entirely by Deep Learning.



7.4 Breast cancer detection

Breast cancer is the second most common cancer in women and men worldwide. In 2012, it represented about 12 percent of all new cancer cases and 25 percent of all cancers in women.by image classification its easy for doctor to diagonosis breast cancer at initial stages



8. CONCLUSION

The majority of this work shows how to implement a CNN which is capable of extracting feature representations from a large amount of labeled data. Next, this work shows how neural network uses binary representation to classify an object into separate classes. Additionally, an attempt is made to optimize the hyperparameter of the CNN to improve the performance. The CNN model presented in this thesis is a very simple deep learning network which effectively extracts useful information for object classification. Adding average pooling to the network helps to simplify further on the calculation and reduces the training time. This proposed network structure can be a valuable baseline for the study of a more advanced deep learning architectures and be used for large-scale image classification tasks. Competitive results are also achieved on the kaggle dogs vs cats dataset. This constitutes an important generalization of deep learning to structured prediction and makes these models suitable for application.

9. REFERENCES :

[1] Antonio Gulli, Sujit Pal. Deep Learning with Keras**:** Get to grips with the basics of Keras to implement fast and efficient deep-learning models.

[2] University Of Montreal : Deep Learning Tutorial.

[3] François Chollet. Deep Learning with Python.

[4] Goodfellow, Bengio, and Courville. Deep Learning. An introduction to a broad range of topics in deep learning, covering mathematical and conceptual background, deep learning techniques used in industry, and research perspectives.

[5] Nikhil Buduma, Nicholas Locascio. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms: With the reinvigoration of neural networks in the 2000s, deep learning has become an extremely active area of research, one that’s paving the way for modern machine learning.

[6] https://docs.anaconda.com/anaconda/- Introduction of Anaconda Navigator and Juypter Labs.

[7] https://www.geeksforgeeks.org/python-programming-language/- Python tutorial