JIMS ENGINEERING AND MANAGEMENT TECHNICAL CAMPUS

GREATER NOIDA (UP)

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY

ACADEMIC SESSION: 2019-20

SUMMER INTERNSHIP REPORT

NAME: HARDIK SETH

BRANCH: ECE

SEMESTER: 5th

ENROLMENT NO. : 00125502817

INDUSTRIAL TRAINING REPORT

INDUSTRIAL SUMMER INTERNSHIP AT HCL,TECHNOLOGIES

JULY-AUGUST

Submitted in partial fulfillment of the

Requirements for the award of

Degree of Bachelor of Technology in Electronics and Communication Engineering

Submitted By

HARDIK SETH

00125502817

Under the guidance of

Mr. ABHINAV GUPTA

SUBMITTED TO:

Department of Electronics and Communication Engineering

JIMS ENGINEERING MANAGEMENT TECHNICAL CAMPUS

GREATER NOIDA

NOVEMBER-2019

CONTENT

CERTIFICATE

HCL TECHNOLOGIES LTD. - IOMC
(A subsidary of HCL Technologies Ltd.)
B-39, Sector 1, Noida-201301, U.P. India
Tel: +91 120 4024700 Fax: +91 120 2425833
Registered office: 806 Siddarth, 96, Nehru Place, New Delhi- 110019 India. www.hclisd.com

30th August, 2019

To Whom So Ever It May Concern

This letter is to certify that Mr. Hardik Seth from JIMS Engineering and Management Technical Campus, Greater Noida, underwent four weeks training program with us from 09th July, 2019 to 08th August, 2019.

During this, he worked on project titled 'Machine Learning and Language processing for multiple languages project'. He has completed his internship on 8th August, 2019 under the supervision of Mr. Abhinav Gupta.

With Regards,

Sujeet Lahiri

Human Resources

HCL Technologies Ltd

DECLARATION

I hereby declare that the Industrial Internship Report entitled **INDUSTRIAL SUMMER TRAINING AT NTPC BADARPUR** is an authentic record of my own work as requirements of Industrial Training during the period from **July to August** which is submitted by us to Department of Electronics and Communication Engineering, JIMS Engineering Management Technical Campus, Greater Noida, in partial fulfillment of requirement for the award of the degree of B. Tech in Electronics and Communication Engineering, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition.

The authors attest that permission has been obtained for the use of any copy righted material appearing in the Project report other than brief excerpts requiring only proper acknowledgement in scholarly writing and all such use has been acknowledged.

ACKNOWLEDGEMENT

I would like to thank **HCL, TECHNOLOGIES, NOIDA** for providing me a golden opportunity to work with them. The support and the environment provided to me during my project was more than what anyone would have expected.

I am very grateful to **Mr. ABHINAV GUPTA(IT)** for giving me his valuable time and constructive guidance in Internship. It would not have been possible to complete the project in such short period of time without their kind encouragement and valuable time.

Best Regards

Hardik Seth

ABSTRACT

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information. Python community has developed many modules to help programmers implement machine learning.

We are living in the 'age of data' that is enriched with better computational power and more storage resources. This data or information is increasing day by day, but the real challenge is to make sense of all the data. Businesses & organizations are trying to deal with it by building intelligent systems using the concepts and methodologies from Data science, Data Mining and Machine learning. Among them, machine learning is the most exciting field of computer science.

It would not be wrong if we call machine learning the application and science of algorithms that provides sense to the data.

This report aims how Machine Learning can be implemented using a well-known computer science language Python.

INTRODUCTION

1.1 ABOUT HCL

HCL Technologies Limited is an Indian multinational information technology (IT) service and consulting company headquartered in Noida, Uttar Pradesh. It is a subsidiary of HCL Enterprise. Originally a research and development division of HCL, it emerged as an independent company in 1991 when HCL entered into the software services business.

The company has offices in 42 countries including the United Kingdom, the United States, France, and Germany with a worldwide network of R&D, "innovation labs" and "delivery centers", and 137,000+ employees and its customers include 250 of the Fortune 500 and 650 of the Global 2000 companies.

It operates across sectors including

Aerospace and Defense

Automotive

Banking

Capital Markets

Chemical and Process industries

Energy and Utilities

Healthcare

Hi-Tech

Industrial manufacturing

Consumer Goods

Insurance

Life Sciences

Manufacturing

Media and Entertainment

Mining and Natural Resources

Oil and gas, retail, telecom, and travel, transportation

Logistics & Hospitality.

HCL Technologies is on the Forbes Global 2000 list. It is among the top 20 largest publicly traded companies in India with a market capitalization of $18.7 billion as of May 2017. As of May 2018, the company, along with its subsidiaries, had a consolidated revenue of $7.8 billion.

ABOUT HCL TECHNOLOGIES

HCL Technologies is a next-generation global technology company that helps enterprises reimagine their businesses for the digital age. Our technology products, services and engineering are built on four decades of innovation, with a world-renowned management philosophy, a strong culture of invention and risk-taking, and a relentless focus on customer relationships. With a worldwide network of R&D, innovation labs and delivery centers, and 147,000+ 'Ideapreneurs' working in 44 countries, HCL serves leading enterprises across key industries, including 250 of the Fortune 500 and 650 of the Global 2000. HCL generated consolidated revenues of USD 9.3 billion for 12 Months Ended 30th Sep, 2019.

They offer an integrated portfolio of products, solutions, services, and IP through our Mode 1-2-3 strategy built around Digital, IoT, Cloud, Automation, Cybersecurity, Analytics, Infrastructure Management and Engineering Services, amongst others, to help enterprises reimagine their businesses for the digital age

VISION AND MISSION

Vision

To be recognized as a Global Information Technology and Process Outsourcing leader in "Delivering Business Value" by providing high-quality and cost-effective solutions".

Mission

To be a chosen platform of growth for all stakeholders and a source of inspiration to community.
Enrich our Shareholders. Enable our Customers.
Empower our Employees. Enhance our Partners.

Corporate Values

HCL Systems operates in accordance with the highest commercial and professional standards in all relationships with its stakeholders – Shareholders, Employees, Customers, Suppliers, business partners and the community.
HCL Systems fosters a climate that encourages innovation, entrepreneurial spirit, diligence and team work amongst employees and recognizes employees accordingly.
HCL Systems has a commitment of building long-term and trusted relationships with its customers, based on consistent delivery of quality and cost effective services. We will meet or exceed our Customer expectations on all our engagements.
HCL Systems is committed to the highest standards of excellence in all things – improving work processes, products, services and creating a conducive environment for its people to perform their best.
HCL Systems believes in professionalism in all facets of its operations – open to honest feedback from others and providing constructive feedback to others. It will conduct all its activities with Integrity and transparency.

ACQUISITION

Number	Acquisition date	Company	Business	Country
1	20 February 2008	Capital Stream Inc.	Business process automation	United States
2	16 July 2008	Liberata Financial Service	Business process outsourcing	United Kingdom
3	25 August 2008	Control Point Solutions, Inc.	Telecommunications service	United States
4	15 December 2008	Axon Group Plc.	SAP Consulting	United Kingdom
5	16 July 2009	UCS Group's Enterprise Solutions SAP Practice	SAP	SA
6	1 April 2016	Part of Volvo IT	Information technology	Sweden
7	19 October 2015	Concept to Silicon Systems (C2SiS)	Semi-Conductors	India
8	29 October 2015	PowerObjects	CRM Consulting	United States
9	2016	Geometric Ltd	PLM & Engineering Services	India
10	17 January 2017	Butler America Aerospace	Engineering & Design	United States
11	25 April 2017	Urban Fulfillment Services, LLC	Business Process Outsourcing	United States
12	5 September 2017	ETL Factory Limited (Datawave)	Automation	United Kingdom
13	27 June 2018	H&D International Group	IT and engineering services provider	Germany

1.2 PYTHON

Python is a popular platform used for research and development of production systems. It is a vast language with number of modules, packages and libraries that provides multiple ways of achieving a task. Python and its libraries like NumPy, SciPy, Scikit-Learn, Matplotlib are used in data science and data analysis. They are also extensively used for creating scalable machine learning algorithms. Python implements popular machine learning techniques such as Classification, Regression, Recommendation, and Clustering. Python offers ready-made framework for performing data mining tasks on large volumes of data effectively in lesser time. It includes several implementations achieved through algorithms such as linear regression, logistic regression, Naïve Bayes, k-means, K nearest neighbor, and Random Forest.

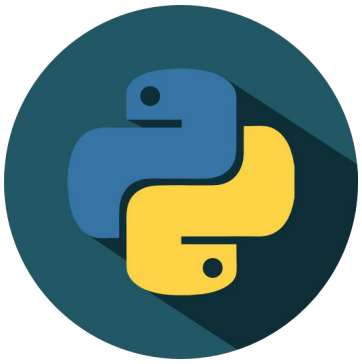


Fig 1.1 PYTHON LOGO

1.3 MACHINE LEARNING

Machine learning is an application of Artificial Intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

1.4 NEED OF MACHINE LEARNING

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

1.5 WHY & WHEN TO MAKE MACHINES LEARN?

We have already discussed the need for machine learning, but another question arises that in what scenarios we must make the machine learn? There can be several circumstances where we need machines to take data-driven decisions with efficiency and at a huge scale.

The followings are some of such circumstances where making machines learn would be more effective.

Lack of human expertise

The very first scenario in which we want a machine to learn and take data-driven decisions, can be the domain where there is a lack of human expertise. The examples can be navigations in unknown territories or spatial planets.

Dynamic scenarios

There are some scenarios which are dynamic in nature i.e. they keep changing over time. In case of these scenarios and behavior, we want a machine to learn and take data-driven decisions. Some of the examples can be network connectivity and availability of infrastructure in an organization.

Difficulty in translating expertise into computational tasks

There can be various domains in which humans have their expertise,; however, they are unable to translate this expertise into computational tasks. In such circumstances we want machine learning. The examples can be the domains of speech recognition, cognitive tasks etc.

1.6 MACHINE LEARNING MODEL

Before discussing the machine learning model, we must need to understand the following formal definition of ML given by professor Mitchell −

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

The above definition is basically focusing on three parameters, also the main components of any learning algorithm, namely Task(T), Performance(P) and experience (E). In this context, we can simplify this definition as −

ML is a field of AI consisting of learning algorithms that −

Improve their performance (P)

At executing some task (T)

Over time with experience (E)

Let us discuss them more in detail now −

Task(T)

From the perspective of problem, we may define the task T as the real-world problem to be solved. The problem can be anything like finding best house price in a specific location or to find best marketing strategy etc. On the other hand, if we talk about machine learning, the definition of task is different because it is difficult to solve ML based tasks by conventional programming approach.

A task T is said to be a ML based task when it is based on the process and the system must follow for operating on data points. The examples of ML based tasks are Classification, Regression, Structured annotation, Clustering, Transcription etc.

Experience (E)

As name suggests, it is the knowledge gained from data points provided to the algorithm or model. Once provided with the dataset, the model will run iteratively and will learn some inherent pattern. The learning thus acquired is called experience(E). Making an analogy with human learning, we can think of this situation as in which a human being is learning or gaining some experience from various attributes like situation, relationships etc. Supervised, unsupervised and reinforcement learning are some ways to learn or gain experience. The experience gained by out ML model or algorithm will be used to solve the task T.

Performance (P)

An ML algorithm is supposed to perform task and gain experience with the passage of time. The measure which tells whether ML algorithm is performing as per expectation or not is its performance (P). P is basically a quantitative metric that tells how a model is performing the task, T, using its experience, E. There are many metrics that help to understand the ML performance, such as accuracy score, F1 score, confusion matrix, precision, recall, sensitivity etc.

CONCEPT OF DEEP LEARNING

2.1 What is Deep Learning ?

Deep learning (also known as **deep structured learning** or **hierarchical learning**) is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, semi-supervised or unsupervised.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

Artificial Neural Networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.

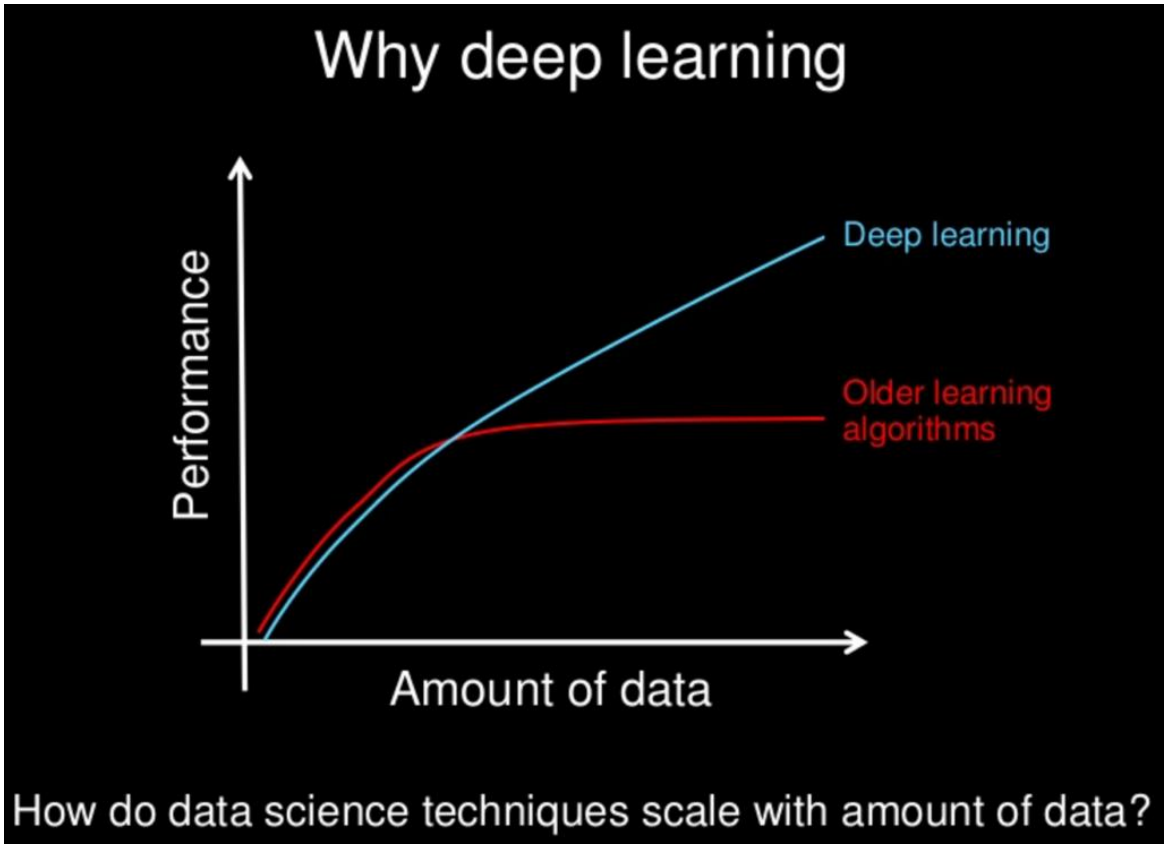


Fig 1.2 : size of data required v/s performance

2.2 What happens inside the neuron?

The input node takes in information in a numerical form. The information is presented as an activation value where each node is given a number. The higher the number, the greater the activation.

Based on the connection strength (weights) and transfer function, the activation value passes to the next node. Each of the nodes sums the activation values that it receives (it calculates the **weighted sum**) and modifies that sum based on its transfer function. Next, it applies an activation function. An activation function is a function that's applied to this particular neuron. From that, the neuron understands if it needs to pass along a signal or not.

Each of the synapses gets assigned weights, which are crucial to **Artificial Neural Networks** (ANNs). Weights are how ANNs learn. By adjusting the weights, the ANN decides to what extent signals get passed along. When you're training your network, you're deciding how the weights are adjusted.
The activation runs through the network until it reaches the output nodes. The output nodes then give us the information in a way that we can understand. Your network will use a cost function to compare the output and the actual expected output. The model performance is evaluated by the cost function. It's expressed as the difference between the actual value and the predicted value. There are many different cost functions you can use, you're looking at what the error you have in your network is. You're working to minimize loss function. (In essence, the lower the loss function, the closer it is to your desired output). The information goes back, and the neural network begins to learn with the goal of minimizing the cost function by tweaking the weights. This process is called **back propagation**.

In **forward propagation**, information is entered into the input layer and propagates forward through the network to get our output values. We compare the values to our expected results. Next, we calculate the errors and propagate the info backward. This allows us to train the network and update the weights. (Backpropagation allows us to adjust all the weights simultaneously.) During this process, because of the way the algorithm is structured, you're able to adjust all of the weights simultaneously. This allows you to see which part of the error each of your weights in the neural network is responsible for.

When you've adjusted the weights to the optimal level, you're ready to proceed to the testing phase!

2.3 How does an artificial neural network learn?

There are two different approaches to get a program to do what you want. First, there's the specifically guided and hard-programmed approach. You tell the program exactly what you want it to do. Then there are **neural networks**. In neural networks, you tell your network the inputs and what you want for the outputs, and then you let it learn on its own.

By allowing the network to learn on its own, you can avoid the necessity of entering in all of the rules. You can create the architecture and then let it go and learn. Once it's trained up, you can give it a new image and it will be able to distinguish output.

Feedforward and feedback networks

A **feedforward** network is a network that contains inputs, outputs, and hidden layers. The signals can only travel in one direction (forward). Input data passes into a layer where calculations are performed. Each processing element computes based upon the weighted sum of its inputs. The new values become the new input values that feed the next layer (feed-forward). This continues through all the layers and determines the output. Feedforward networks are often used in, for example, data mining.

A **feedback network** (for example, a recurrent neural network) has feedback paths. This means that they can have signals traveling in both directions using loops. All possible connections between neurons are allowed. Since loops are present in this type of network, it becomes a non-linear dynamic system which changes continuously until it reaches a state of equilibrium. Feedback networks are often used in optimization problems where the network looks for the best arrangement of interconnected factors.

2.4 What is a weighted sum?

Inputs to a neuron can either be features from a training set or outputs from the neurons of a previous layer. Each connection between two neurons has a unique synapse with a unique weight attached. If you want to get from one neuron to the next, you have to travel along the synapse and pay the "toll" (weight). The neuron then applies an activation function to the sum of the weighted inputs from each incoming synapse. It passes the result on to all the neurons in the next layer. When we talk about updating weights in a network, we're talking about adjusting the weights on these synapses.

Stochastic Gradient Descent
A neuron's input is the sum of weighted outputs from all the neurons in the previous layer. Each input is multiplied by the weight associated with the synapse connecting the input to the current neuron. If there are 3 inputs or neurons in the previous layer, each neuron in the current layer will have 3 distinct weights: one for each synapse.

In a nutshell, the activation function of a node defines the output of that node.
The activation function (or transfer function) translates the input signals to output signals. It maps the output values on a range like 0 to 1 or -1 to 1. It's an abstraction that represents the rate of action potential firing in the cell. It's a number that represents the likelihood that the cell will fire. At it's simplest, the function is binary: **yes**(the neuron fires) or **no**(the neuron doesn't fire). The output can be either 0 or 1 (on/off or yes/no), or it can be anywhere in a range. If you were using a function that maps a range between 0 and 1 to determine the likelihood that an image is a cat, for example, an output of 0.9 would show a 90% probability that your image is, in fact, a cat.

2.5 What is an activation function?

In a nutshell, the activation function of a node defines the output of that node.
The activation function (or transfer function) translates the input signals to output signals. It maps the output values on a range like 0 to 1 or -1 to 1. It's an abstraction that represents the rate of action potential firing in the cell. It's a number that represents the likelihood that the cell will fire. At it's simplest, the function is binary: **yes** (the neuron fires) or **no** (the neuron doesn't fire). The output can be either 0 or 1 (on/off or yes/no), or it can be anywhere in a range.

PROJECT

"IMAGE AND LANGUAGE PROCESSING ON MULTIPLE LANGUAGES"

What does Program do?

The deep learning program performs 4 task

A. Object Detection

B. OCR (Optical Character Recognition)

C. Qr and Bar code scan

D. Language Translation

What do we need?

Well, since we are making this program in Python we need Python 3 installed on our computer, run the installer and once python is installed we can proceed to install other dependencies.

Dependencies

Scikit-learn

NumPy

SciPy

Keras

Pyzbar

Pytesseract

Tkinter

Run"CMD", this will open the command prompt and type in the following commands.

pip install NumPy

pip install Scikit-learn

Pip install keras

Pip install pyzbar

Pip install pytesseract

The project had many files and each file had code for different task

File 1 : Training Object Detection python file

CODE

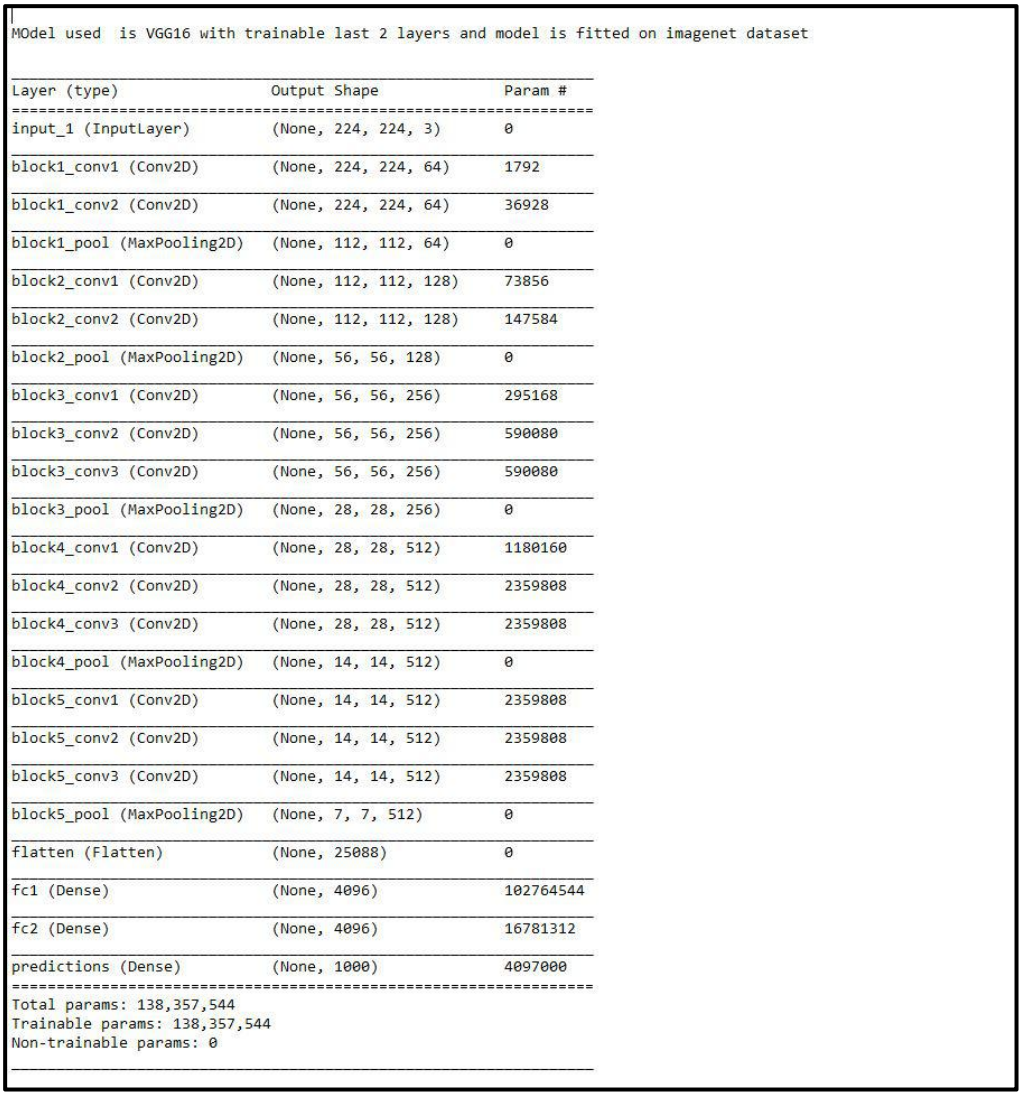
```
#-------------------------library used-----------------------------------------------------
import numpy as np
from keras.utils import np_utils
from sklearn.utils import shuffle
import cv2
import os
import glob
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from keras.preprocessing import image
from keras.applications.vgg16 import VGG16
from keras.models import Model
from keras.layers import Dense,Flatten
def model_train():
#------------------------data loading and preprocessing-----------------------------------
        no_classes=9
        epochs=40
        X,y=[],[]
        train_data_dir="C:\\Users\\hseth\\Desktop\\HCL_project_1\\data\\"

        train_data=[train_data_dir+i for i in os.listdir(train_data_dir)]
        for img in train_data:
                im=image.load_img(img,target_size=(224,224))
                image_array=image.img_to_array(im)
```

```
                X.append(image_array)
        for i in train_data:
                if "telephone" in i:
                        y.append(0)
                elif "mug" in i:
                        y.append(1)
                elif "dog" in i:
                        y.append(2)
                elif "cat" in i:
                        y.append(3)
                elif "pen" in i:
                        y.append(4)
                elif "bottle" in i:
                        y.append(5)
                elif "watch" in i:
                        y.append(6)
                elif "headphone" in i:
                        y.append(7)
                elif "book" in i:
                        y.append(8)
#---------------------------------------data-----------------------------------
        X=np.array(X)
        y=np.array(y)
        X,y=shuffle(X,y)
#-------------------------------splitting of data---------------------------------
        Xtrain,Xtest,ytrain,ytest=train_test_split(X,y,test_size=0.2,random_state=0)
#-------------------------------normalizing the images----------------------------
```

```
	Xtrain=Xtrain/255
	Xtest=Xtest/255
	print("[INFO] shape of training data:",Xtrain.shape,"\n[INFO] shape of testing data:",Xtest.shape)

#------------------------------data loaded-----------------------------------------
	cv2.imshow("random Image",Xtrain[56])
	print("image is of class:",ytrain[56])
	cv2.waitKey(0)
	cv2.destroyAllWindows()
#------------------------------making o/p categorical------------------------------
	ytrain=np_utils.to_categorical(ytrain,no_classes)
	ytest=np_utils.to_categorical(ytest,no_classes)
#----------------------------------data augmentation-----------------------------------
	datagen = image.ImageDataGenerator(featurewise_center=True,featurewise_std_normalization=True,rotation_range=20,width_shift_range=0.2,height_shift_range=0.2,horizontal_flip=True)
	datagen.fit(Xtrain)

#------------------------------deep learning model-----------------------------------
	model=VGG16(weights="imagenet",include_top=False,input_shape=(224,224,3))

	x= model.output
	x= Flatten()(x)
	x= Dense(512,activation='relu')(x)
	x= Dense(215,activation='relu')(x)
	pred=Dense(no_classes,activation="softmax")(x)

#creating the final model
```

```
        model=Model(inputs=model.input,outputs=pred)
# setting no. of untrained layers in our final model
        for layer in model.layers[:-3]:
                layer.trainable = False
#----------------------------------COMPILING THE MODEL---------------------------------
        model.compile(loss="categorical_crossentropy",optimizer="Adadelta",metrics=['accuracy'])
#----------------------------------FITTING THE MODEL-----------------------------------
        lit_data=datagen.flow(Xtrain, ytrain, batch_size=1)
        model.fit_generator(lit_data,steps_per_epoch=len(X)/12,epochs=epochs,verbose=1,validation_data=(Xtest,ytest))


#----------------------------------SAVING MODEL----------------------------------------------
        print("[INFO] Model Saved at Location:C:\\Users\\hseth\\Desktop\\HCL_project_1\\ as model_object_detect.h5")
        model.save("C:\\Users\\hseth\\Desktop\\HCL_project_1\\model_object_detect.h5")
```

Fig 3.1: example of dataset used



```
MOdel used  is VGG16 with trainable last 2 layers and model is fitted on imagenet dataset

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 25088)             0
_________________________________________________________________
fc1 (Dense)                  (None, 4096)              102764544
_________________________________________________________________
fc2 (Dense)                  (None, 4096)              16781312
_________________________________________________________________
predictions (Dense)          (None, 1000)              4097000
=================================================================
Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0
_________________________________________________________________
```

Fig 3.2: VGG16 architecture

This file contains code for training the deep learning model for Object detection. The datatset is self created having 40 photos of each object that needs to be detected.

Due to small size of datatset I used the concept of DATA AUGMENTATION.

This method increases the size of dataset by changing some of the aspects of the image.

Rather than scratch coding CNN neural network I used the concept of TRANSFER LEARNING.

In this I basically downloaded the pretrained VGG16 model from keras and untrained the last 3 layers and trained them on my own dataset.

Why I used VGG16 over much accurate models available online?

The reason behind this is that VGG16 is very light compared to other models available.

Why I didn't used whole model trained on IMAGENET dataset? No scope of learning if use direct model and also model took time to produce prediction.(Time Complexity)

FILE 2: LANGUAGE TRANSLATION SCRATCH FOR 1 LANGUAGE

CODE :

```
import numpy as np
from keras.preprocessing.sequence import pad_sequences
from keras.preprocessing.text import Tokenizer
from keras.utils import np_utils
from keras.models import Model
from keras.layers import Input,LSTM,Dense,GRU,Flatten
from keras.layers import Embedding
from sklearn.model_selection import train_test_split
import tensorflow as tf
#--------------------------Accessing the Dataset given to us and splitting it ----------------------------------------


def sparse_cross_entropy(y_true, y_pred):
    
    loss = tf.nn.sparse_softmax_cross_entropy_with_logits(labels=y_true,logits=y_pred)
    loss_mean = tf.reduce_mean(loss)

    return loss_mean
eng_text=[]
spanish_text=[]
data_size=123013
start_char = "ssss"        #start char
end_char = "eeee"          #end char
with open("/home/hardik/Desktop/spa.txt","r",encoding="utf-8") as file:
    lines=file.read().split("\n")
```

```
for line in lines[:min(data_size,len(lines)-1)]:
    input_texts,target_texts=line.split("\t")
    target_texts=start_char+" " + target_texts +" "+ end_char
    eng_text.append(input_texts)
    spanish_text.append(target_texts)
# printing random text and its translation
print("[INFO] Example Text in English:",eng_text[12034])
print("[INFO] Example Translation in Spanish:",spanish_text[12034])

eng_tokenizer=Tokenizer()
eng_tokenizer.fit_on_texts(eng_text)
eng_length=max(len(line.split()) for line in eng_text)
eng_vocab_size=len(eng_tokenizer.word_counts)+1

spa_tokenizer=Tokenizer()
spa_tokenizer.fit_on_texts(spanish_text)
spa_length=max(len(line.split()) for line in spanish_text)
spa_vocab_size=len(spa_tokenizer.word_counts)+1
print("[INFO]English vocab size: ",eng_vocab_size)
print("[INFO]Max Length of the English data:",eng_length)
print("[INFO]Spanish vocab size: ",spa_vocab_size)
print("[INFO]Max Length of the spanish data:",spa_length)

data_input=eng_tokenizer.texts_to_sequences(eng_text)
data_input=[list(reversed(x)) for x in data_input]
data_input=pad_sequences(data_input,maxlen=eng_length,padding="pre")
```

```
data_output=spa_tokenizer.texts_to_sequences(spanish_text)
data_output=pad_sequences(data_output,maxlen=spa_length,padding="post")
validation_split = 1000 / len(data_input)
validation_split

decoder_input_data=data_output[:,:-1]
decoder_output_data=data_output[:,1:]

encoder_input=Input(shape=(None, ))
net=Embedding(input_dim=eng_vocab_size,output_dim=128)(encoder_input)
net=LSTM(512,return_sequences=True)(net)
net=LSTM(512,return_sequences=True)(net)
net=LSTM(512,return_sequences=False)(net)
encoder_output=net


decoder_input=Input(shape=(None, ))
net2=Embedding(input_dim=spa_vocab_size,output_dim=128)(decoder_input)
net2=GRU(512,return_sequences=True)(net2,initial_state=encoder_output)
net2=GRU(512,return_sequences=True)(net2,initial_state=encoder_output)
net2=GRU(512,return_sequences=True)(net2,initial_state=encoder_output)

decoder_output=Dense(spa_vocab_size,activation="linear")(net2)

decoder_target = tf.placeholder(dtype='int32', shape=(None, None))
```

```
model=Model(inputs=[encoder_input,decoder_input],outputs=[decoder_output])


model.compile(optimizer="RMSprop",loss=sparse_cross_entropy,target_tensors=[decoder_target])

print(model.summary())

model.fit(x=[data_input,decoder_input_data],y=decoder_output_data,batch_size=125,epochs=1,validation_split
=validation_split)

model.save("language_trans.h5")
```

Now that the weather is warmer, I can go outdoors.	Ahora que el tiempo es cálido, puedo salir fuera.
Nowadays there are no ninjas or samurais in Japan.	Hoy en día no hay ninjas o samuráis en Japón.
Of course, a license is needed to operate a crane.	Por supuesto que se requiere licencia para operar una grúa.
One of my favorite tunes was playing on the radio.	Una de mis canciones favoritas estaba sonando en la radio.
Only one little boy survived the traffic accident.	Solo un niño pequeño sobrevivió al accidente de tránsito.
Our English teacher put emphasis on pronunciation.	Nuestro profesor de inglés le pone énfasis a la pronunciación.
Our bike tour of the French Alps lasted two weeks.	Nuestro tour en bicicleta por los alpes franceses duró dos semanas.
Owing to the storm, the ship could not leave port.	El barco no pudo dejar el puerto debido a la tormenta.
People say I look about the same age as my sister.	La gente dice que luzco casi de la misma edad que mi hermana.
People say I look about the same age as my sister.	La gente dice que parezco más o menos de la misma edad que mi hermana.
Perhaps Tom has already talked to Mary about that.	Puede que Tom haya hablado ya de eso con Mary.
Personally, I don't think we should be doing this.	Personalmente, pienso que no deberíamos estar haciendo esto.
Please be sure to take one dose three times a day.	Por favor, asegúrese de tomar una dosis tres veces al día.
Please clean up this room before everyone arrives.	Por favor limpia esta habitación antes de que todo el mundo llegue.
Please come over for lunch the day after tomorrow.	Por favor, ven a almorzar a mi casa pasado mañana.
Please don't turn up the volume on the television.	Por favor, no suba el volumen del televisor.
Please turn in the report by the end of the month.	Por favor, entreguen el reporte para el final del mes.
Please turn off the light when you leave the room.	Apaga la luz cuando salgas de la pieza, por favor.
President Truman had to make a difficult decision.	El presidente Truman tuvo que tomar una difícil decisión.
Press the red button if something strange happens.	Pulsa el botón rojo si pasa algo raro.
Recreational drug use inspires many urban legends.	El consumo de drogas recreativas inspira muchas leyendas urbanas.
Revolutions that don't succeed are soon forgotten.	Las revoluciones que fracasan son rápidamente olvidadas.
Scientists are working hard to put an end to AIDS.	Los científicos están trabajando duro para ponerle fin al SIDA.
Scientists consider the invention to be important.	Científicos lo consideran un invento importante.
She advised him to talk about his life in America.	Ella le aconsejó que hablara acerca de su vida en Estados Unidos.
She always lets her children do what they want to.	Ella siempre deja a sus hijos hacer lo que quieran.
She always wears clothes which are out of fashion.	Ella siempre usa pilchas pasadas de moda.
She asked him to call her later, but he forgot to.	Ella le pidió que le llamara más tarde, pero a él se le olvidó.
She asked him to help her father clean the garage.	Le pidió que le ayudara a su padre a limpiar la cochera.
She came to Tokyo when she was eighteen years old.	Ella vino a Tokio cuando tenía dieciocho años.
She can play the violin, not to mention the piano.	Ella puede tocar el violín, y ni mencionar el piano.

Fig 4: datatset for language translation Spanish to English

As you can clearly see the code for langauge translation from one language to another is a hectic job and even takes lots of time to train therefore to make code do language translation over any language I had to use an API provided by google “googletrans” .

This API made code of language translation confined to just 10 to 15 lines of code.

```
import tkinter as tk

from tkinter import filedialog

from googletrans import Translator

from os import system
```

```
from webcam import UploadAction,pick_file


def translate():
    translator = Translator()
    print("ENTER 1:TO ENTER TEXT YOU WANT OT TRANSLATE\nENTER 2:TO UPLOAD TEXT FILE YOU WANT TO TRANSLATE")
    char=int(input("\nEnter your Choice:"))
    if char is 1:
        text=input("Type Text you want to Translate:")
        trans=translator.translate(text)
        print("\n\nLanguage you Entered Text is--------->",trans.src,"\n\nTranslation of above Text in English is:\n",trans.text)
    elif char is 2:
        system("clear")
        filename=pick_file()
        file_up_name=input("Enter Name of file you want to save data as:")
        with open(filename,"r",encoding="utf-8") as file:
            text=file.read()
            trans=translator.translate(text)
            print("\n\nLanguage you Entered Text is--------->",trans.src)
        file_up="C:\\Users\\hseth\\Desktop\\"+file_up_name+".txt"
        print(file_up)
        file_des=open(file_up,"w",encoding="utf-8")
        file_des.write(trans.text)
        file_des.close()
```

This code can do language translation for all languages without requirement of training it.

This approch of language translation is easy and can be used for creating a application also.

FILE 3: OCR (OPTICAL CHARACTER RECOGNITION)

CODE:

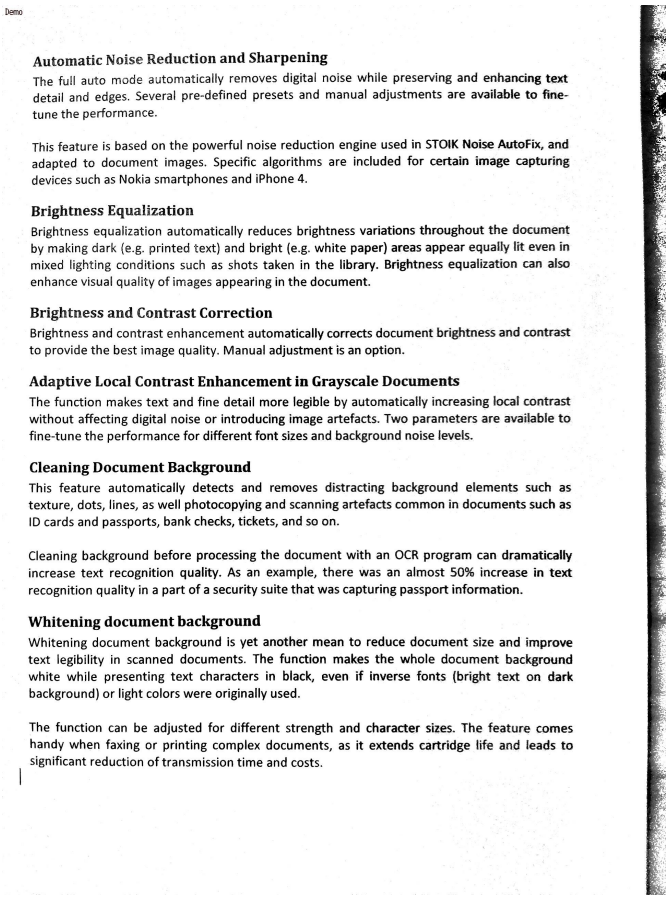
```
import pytesseract

def ocr_doc(img):

    text = pytesseract.image_to_string(img)

    return text
```

Pytessearct is again an API provided by google.

Optical character recognition is one of the most toughest problem I have ever tried to solve. Due to lack of time and no availability of dataset I had to use an API to tackle this aspect of the task.

The above code did reads text from the image and copy the text in .text file if required

Filename can be given by the user.



Demo

Automatic Noise Reduction and Sharpening

The full auto mode automatically removes digital noise while preserving and enhancing text detail and edges. Several pre-defined presets and manual adjustments are available to fine-tune the performance.

This feature is based on the powerful noise reduction engine used in STOIK Noise AutoFix, and adapted to document images. Specific algorithms are included for certain image capturing devices such as Nokia smartphones and iPhone 4.

Brightness Equalization

Brightness equalization automatically reduces brightness variations throughout the document by making dark (e.g. printed text) and bright (e.g. white paper) areas appear equally lit even in mixed lighting conditions such as shots taken in the library. Brightness equalization can also enhance visual quality of images appearing in the document.

Brightness and Contrast Correction

Brightness and contrast enhancement automatically corrects document brightness and contrast to provide the best image quality. Manual adjustment is an option.

Adaptive Local Contrast Enhancement in Grayscale Documents

The function makes text and fine detail more legible by automatically increasing local contrast without affecting digital noise or introducing image artefacts. Two parameters are available to fine-tune the performance for different font sizes and background noise levels.

Cleaning Document Background

This feature automatically detects and removes distracting background elements such as texture, dots, lines, as well photocopying and scanning artefacts common in documents such as ID cards and passports, bank checks, tickets, and so on.

Cleaning background before processing the document with an OCR program can dramatically increase text recognition quality. As an example, there was an almost 50% increase in text recognition quality in a part of a security suite that was capturing passport information.

Whitening document background

Whitening document background is yet another mean to reduce document size and improve text legibility in scanned documents. The function makes the whole document background white while presenting text characters in black, even if inverse fonts (bright text on dark background) or light colors were originally used.

The function can be adjusted for different strength and character sizes. The feature comes handy when faxing or printing complex documents, as it extends cartridge life and leads to significant reduction of transmission time and costs.

Fig 5: test file for OCR code

FILE 4: Qr and Bar Code scan

CODE:

```
import pyzbar.pyzbar as pyzbar
import numpy as np
def decode(im) :
  # Find barcodes and QR codes
  decodedObjects = pyzbar.decode(im)
  for obj in decodedObjects:
    print('Type : ', obj.type)
    print('Data : ', obj.data,'\n')
```

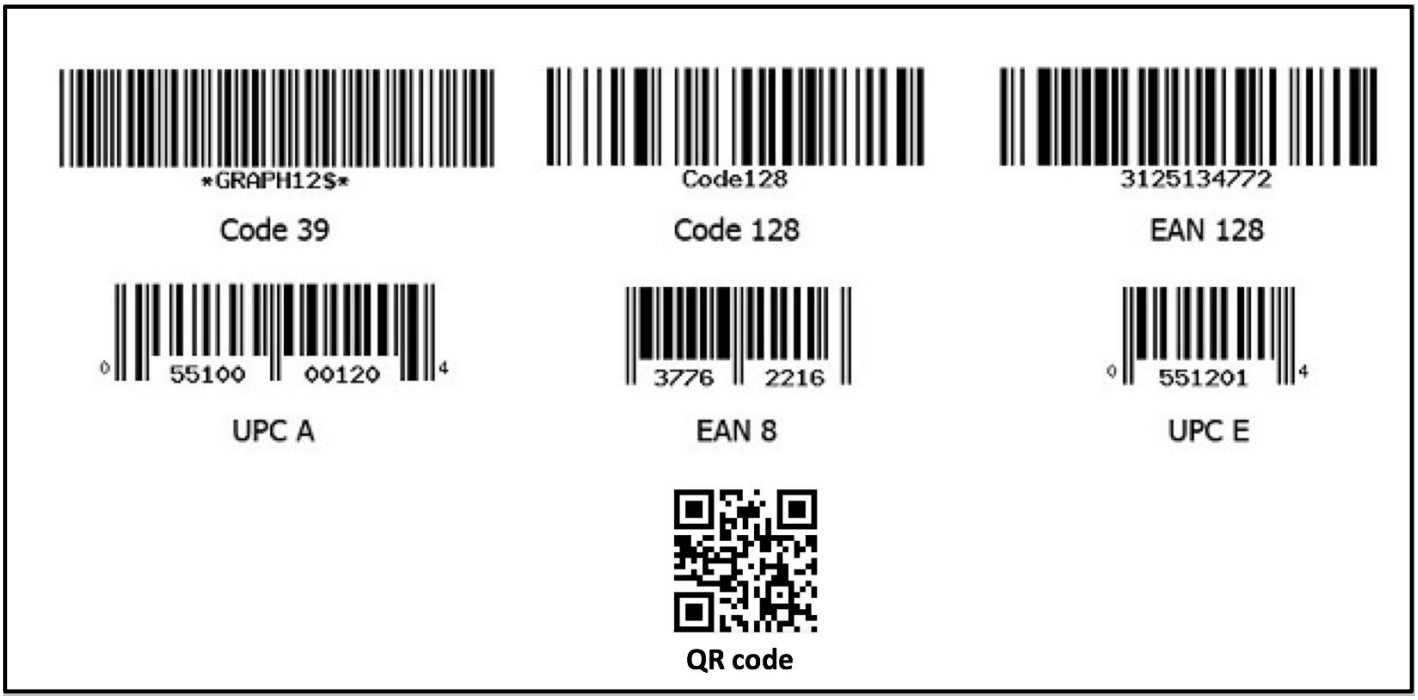


Fig 6: test image for qr code scan

Fig 7: output of the QR code scan file

FILE 5: ALL INPUT FUNCTIONS

CODE:

```
import numpy as np
from keras.preprocessing.image import load_img, img_to_array
from keras.applications.vgg16 import preprocess_input
import cv2
import tkinter as tk
from tkinter import filedialog


def UploadAction(event=None):
    filename = filedialog.askopenfilename()
    print('Selected:', filename)
    return filename


def pick_file():
        root = tk.Tk()
        button = tk.Button(root, text='Open', command=UploadAction)
        button.pack()
        filename=UploadAction()
        return filename
def camera():
        camera = cv2.VideoCapture(0)
        print("enter *s* to save the image ")
        while True:
                return_value,image = camera.read()
                gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
                cv2.imshow('image',gray)
```

```
                if cv2.waitKey(1)& 0xFF == ord('s'):
                        cv2.imwrite("C:\\Users\\hseth\\Desktop\\HCL_project_1\\test.jpg",image)
                        break
        camera.release()
        cv2.destroyAllWindows()
        img=image_get_normal(2)
        return img
def image_get_object_detect():
        print("Select the file:")
        input_image_dir=pick_file()
        image = load_img(input_image_dir, target_size=(224, 224))
        return image
def image_preprocess(image):
        image = img_to_array(image)
        image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))

        image = preprocess_input(image)
        return image
def image_get_normal(value):
        if value is 2:
                input_image_dir="C:\\Users\\hseth\\Desktop\\HCL_project_1\\test.jpg"
                image = load_img(input_image_dir, target_size=(224, 224))
        elif value is 0:
                print("Select the file:")
                input_image_dir=pick_file()
                image=cv2.imread(input_image_dir,0)
        else:
```

```
                print("Select the file:")
                input_image_dir=pick_file()
                image = cv2.imread(input_image_dir)

        return image
```

The above code contains all the input methods used to get data in the code.

Creating seperate file made the task of getting input easy and readable.

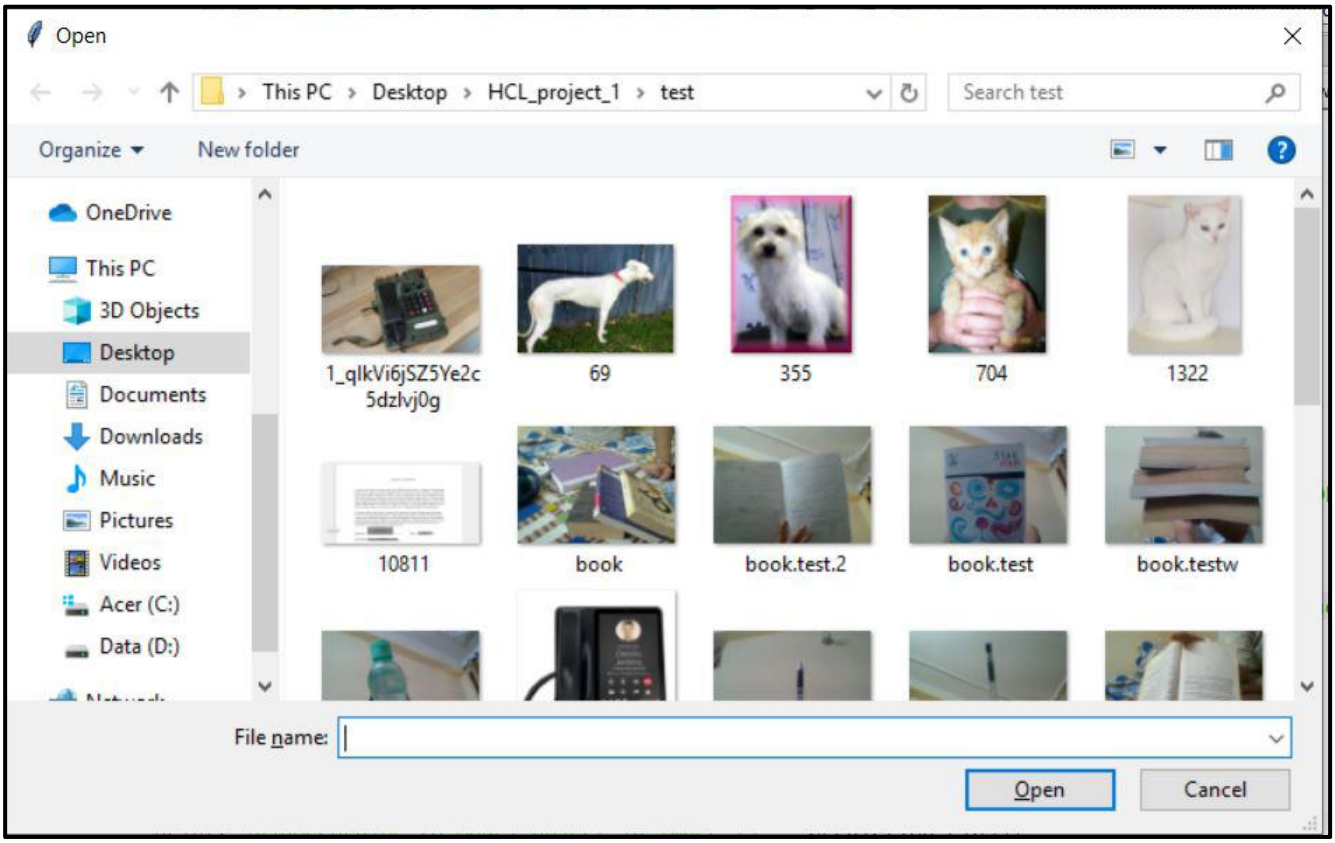


Fig 8: one of the way to give input to the image

FILE 6: OUTPUT FUNCTION FOR OBJECT DETECTION

CODE:

```
from keras.models import load_model
from os import system
def model_pred(image):
        model = load_model("C:\\Users\\hseth\\Desktop\\HCL_project_1\\model_object_detect.h5")
        yhat = model.predict(image)
        system("clear")
        print("[info] predicted probabilities are:",yhat)
        for i in range(9):
                if yhat[0][i]>=0.5:
                        yhat[0][i]=1
                elif yhat[0][i]<0.5:
                        yhat[0][i]=0
        for i in range(9):
                if yhat[0][i]==1:
                        loc=i
        if loc == 0:
                class_name="Telephone"
        elif loc == 1:
                class_name="Mug"
        elif loc == 2:
                class_name="Dog"
        elif loc == 3:
                class_name="Cat"
        elif loc == 4:
                class_name="Pen"
```

```
        elif loc == 5:
                class_name="Bottle"
        elif loc == 6:
                class_name="Watch"
        elif loc == 7:
                class_name="Headphone"
        elif loc == 8:
                class_name="Book"
        return class_name
```

Fig 9: example of how output is shown in object detection

FILE 7: DISPLAY AND OUTPUT FILE

CODE:

```
from qr_code import decode

from obj_det_scr_op import model_pred

from ocr import ocr_doc

from object_detection_scratch import model_train

from webcam import camera,image_get_object_detect,image_get_normal,image_preprocess

import numpy as np

from os import system

import time

import sys

from lag_using_google import translate

back="yes"

while back == "yes":

        system("clear")

        print("ENTER 1:OBJECT DETECTION\nENTER 2:OPTICAL CHARACTER
RECOGINITION\nENTER 3:QR AND BARCODE DECODER\nENTER 4:LANGUAGE
TRANSLATION\nENTER 5:TRAIN THE OBJECT DETECTION MODEL\nENTER 6:TO EXIT")

        choice=input("\n\nEnter Your Choice:")

        system("clear")

        if choice =="1" :

                print("The Model can predict following
objects :\n1.Dog\n2.Cat\n3.Watch\n4.Pen\n5.Mug\n6.Headphones\n7.Book\n8.Bottle\n9.Telephone\n")

                print("ENTER 1:TO USE IMAGE LOCATION\nENTER 2:USE WEBCAM TO TAKE
PICTURE")

                choice_obj=input("\n\nEnter Your Choice:")

                system("clear")

                if choice_obj == "1":
```

```
                image=image_get_object_detect()
                pre_image=image_preprocess(image)
                prediction_class=model_pred(pre_image)
                print("\n\nAccording to Model Object in Image is :",prediction_class)
                back=input("\n\n would you like to go back(yes/no):")
                if back=="no":
                        print("EXITING")
                        time.sleep(3)
                        sys.exit()
        elif choice_obj == "2":
                image=camera()
                pre_image=image_preprocess(image)
                prediction_class=model_pred(pre_image)
                print("\n\nAccording to Model Object in Image is :",prediction_class)
                back=input("\n\n would you like to go back(yes/no):")
                if back=="no":
                        print("EXITING")
                        time.sleep(3)
                        sys.exit()

        else :
                print("Wrong input.....")
                back=input("\n\n would you like to go back(yes/no):")
                if back=="no":
                        print("EXITING")
                        time.sleep(3)
                        sys.exit()
```

```
        elif choice == "2":
                image=image_get_normal(0)
                text=ocr_doc(image)
                system("clear")
                print("The Text in the Picture is :\n",text)
                print("--------------------------------------------------------------------------------------------------------------------")
                print("\n\nENTER 1:TO SAVE THE DOCUMENT IN TEXT FILE\nENTER ANY OTHER KEY TO EXIT WITHOUT SAVING")
                ch=input("\n\nENTER YOUR CHOICE:")
                system("clear")
                if ch == "1":
                        file_up_name=input("Enter Name of file you want to save data as:")
                        file_up="/home/hardik/Desktop/"+file_up_name+".txt"
                        print("File Stored at Location:",file_up)
                        file_des=open(file_up,"w",encoding="utf-8")
                        file_des.write(text)
                        file_des.close()
                        back=input("\n\n would you like to go back(yes/no):")
                        if back=="no":
                                print("EXITING")
                                time.sleep(3)
                                sys.exit()
                else:
                        back=input("\n\n would you like to go back(yes/no):")
                        if back=="no":
                                print("EXITING")
                                time.sleep(3)
```

```
                                    sys.exit()
        elif choice == "3":
                    image=image_get_normal(1)
                    decode(image)
                    back=input("\n\n would you like to go back(yes/no):")
                    if back=="no":
                                print("EXITING")
                                time.sleep(3)
                                sys.exit()
        elif choice == "4":
                    translate()
                    back=input("\n\n would you like to go back(yes/no):")
                    if back=="no":
                                print("EXITING")
                                time.sleep(3)
                                sys.exit()
        elif choice == "5":
                    model_train()
        elif choice == "6":
                    print("EXITING.....")
                    time.sleep(3)
                    sys.exit()
        else:
                    print("WRONG INPUT!!!!")
                    back=input("\n\n would you like to go back(yes/no):")
                    if back=="no":
                                print("EXITING")
```

```
                    time.sleep(3)

                    sys.exit()
```

```
 would you like to go back(yes/no):yes
ENTER 1:OBJECT DETECTION
ENTER 2:OPTICAL CHARACTER RECOGINITION
ENTER 3:QR AND BARCODE DECODER
ENTER 4:LANGUAGE TRANSLATION
ENTER 5:TRAIN THE OBJECT DETECTION MODEL
ENTER 6:TO EXIT



Enter Your Choice:4
ENTER 1:TO ENTER TEXT YOU WANT OT TRANSLATE
ENTER 2:TO UPLOAD TEXT FILE YOU WANT TO TRANSLATE


Enter your Choice:
```

Fig 10 : 1st page of the code generated by above code

```
ENTER 1:OBJECT DETECTION
ENTER 2:OPTICAL CHARACTER RECOGINITION
ENTER 3:QR AND BARCODE DECODER
ENTER 4:LANGUAGE TRANSLATION
ENTER 5:TRAIN THE OBJECT DETECTION MODEL
ENTER 6:TO EXIT



Enter Your Choice:5
C:\Users\hseth\Anaconda3\envs\tf_gpu\lib\site-packages\PIL\Image.py:989: UserWarning: Palette images with Transparency expressed in bytes should be
converted to RGBA images
  "Palette images with Transparency expressed in bytes should be "
[INFO] shape of training data: (323, 224, 224, 3)
[INFO] shape of testing data: (81, 224, 224, 3)
image is of class: 4
```

Fig 11: Training of Object Detection

```
34/33 [==============================] - 5s 140ms/step - loss: 0.6284 - accuracy: 0.8235 - val_loss: 1.2629 - val_accuracy: 0.5556
Epoch 20/30
34/33 [==============================] - 5s 140ms/step - loss: 0.3821 - accuracy: 0.9118 - val_loss: 1.6249 - val_accuracy: 0.5062
Epoch 21/30
34/33 [==============================] - 5s 140ms/step - loss: 0.3552 - accuracy: 0.8529 - val_loss: 1.5454 - val_accuracy: 0.5556
Epoch 22/30
34/33 [==============================] - 5s 140ms/step - loss: 0.4384 - accuracy: 0.8529 - val_loss: 1.1644 - val_accuracy: 0.6173
Epoch 23/30
34/33 [==============================] - 5s 141ms/step - loss: 0.4695 - accuracy: 0.8235 - val_loss: 2.0166 - val_accuracy: 0.4198
Epoch 24/30
34/33 [==============================] - 5s 141ms/step - loss: 0.1999 - accuracy: 0.9412 - val_loss: 1.0132 - val_accuracy: 0.5926
Epoch 25/30
34/33 [==============================] - 5s 141ms/step - loss: 0.7762 - accuracy: 0.7647 - val_loss: 1.5895 - val_accuracy: 0.5556
Epoch 26/30
34/33 [==============================] - 5s 143ms/step - loss: 0.5880 - accuracy: 0.8235 - val_loss: 0.9733 - val_accuracy: 0.5926
Epoch 27/30
34/33 [==============================] - 5s 141ms/step - loss: 0.5901 - accuracy: 0.8235 - val_loss: 1.2063 - val_accuracy: 0.6173
Epoch 28/30
34/33 [==============================] - 5s 141ms/step - loss: 0.4300 - accuracy: 0.8235 - val_loss: 1.4288 - val_accuracy: 0.4938
Epoch 29/30
34/33 [==============================] - 5s 140ms/step - loss: 0.4109 - accuracy: 0.8529 - val_loss: 2.2755 - val_accuracy: 0.3704
Epoch 30/30
34/33 [==============================] - 5s 140ms/step - loss: 0.2063 - accuracy: 0.9412 - val_loss: 1.0285 - val_accuracy: 0.6173
[INFO] Model Saved at Location:C:\Users\hseth\Desktop\HCL_project_1\ as model_object_detect.h5
```

Fig 12 : End of training

```
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
 4874240/58889256 [=>............................] - ETA: 5:33
```

Fig 13: downloading pretrained VGG16 model

```
ENTER 4:LANGUAGE TRANSLATION
ENTER 5:TRAIN THE OBJECT DETECTION MODEL
ENTER 6:TO EXIT



Enter Your Choice:4
ENTER 1:TO ENTER TEXT YOU WANT OT TRANSLATE
ENTER 2:TO UPLOAD TEXT FILE YOU WANT TO TRANSLATE


Enter your Choice:2
Selected: C:/Users/hseth/Desktop/HCL_project_1/test/trans_test

Enter Name of file you want to save data as:new_trans


Language you Entered Text is---------> es
C:\Users\hseth\Desktop\new_trans.txt



 would you like to go back(yes/no):yes
ENTER 1:OBJECT DETECTION
ENTER 2:OPTICAL CHARACTER RECOGINITION
ENTER 3:QR AND BARCODE DECODER
ENTER 4:LANGUAGE TRANSLATION
ENTER 5:TRAIN THE OBJECT DETECTION MODEL
ENTER 6:TO EXIT



Enter Your Choice:
```

Fig 14: language translation output

CONCLUSION

Machine Learning is simply making a computer perform a task without explicitly programming it. In today's world every system that does well has a machine learning algorithm at its heart. Take for example Google Search engine, Amazon Product recommendations, LinkedIn, Facebook etc, all these systems have machine learning algorithms embedded in their systems in one form or the other.

They are efficiently utilising data collected from various channels which helps them get a bigger picture of what they are doing and what they should do.

Python is a widely used high-level programming language for general-purpose programming. Apart from being open source programming language, python is a great object-oriented, interpreted, and interactive programming language. Python combines remarkable power with very clear syntax.

Finally, when it comes to the development of machine learning models of your own, you looked at the choices of various development languages, IDEs and Platforms.

REFERENCES

1. https://towardsdatascience.com/creating-your-own-object-detector-ad69dda69c85
2. https://towardsdatascience.com/language-translation-with-rnns-d84d43b40571
3. CODING ELEMENTS
4. https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/
5. https://github.com/sharkdp/qrnet