Airplane Image Classification using Keras

(CNN - Deep Learning)

For the fulfillment of 7th Semester

Training conducted at:

INSTITUTE OF SYSTEM STUDY AND ANALYSES (ISSA)

Defence R&D Organization (DRDO),

Ministry of Defence,

Metcalfe House, Delhi-110054

Submitted to:	**Submitted By:**
Mr. Babloo Saha	**Deepika Ghotra**
(Scientist 'D')	**MRU -CSE**

ACKNOWLEDGMENT

First and foremost, I would like to express my deep sense of gratitude to **Mr. Anurag Pathak, Head HRD,** INSTITUTE FOR SYSTEM STUDIES AND ANALYSIS (ISSA), DRDO, New Delhi, for giving me the opportunity to work as a trainee in his esteemed organization and providing all the necessary resources and expertise for this purpose.

I am highly grateful to **Mr. BABLOO SAHA, Scientist-'D',** for assigning me an innovative project and for their valuable guidance. I would definitely like to place on record my thanks for their constant interest, encouragement, motivation, and support, without which this project would not have taken its present shape.

I sincerely thank **Mr. BABLOO SAHA, Scientist-'D'**, ISSA, for his valuable guidance and important contributions throughout the life cycle of the project.

DECLARATION

I hereby declare that the Industrial Training Report entitled "**AIRPLANE IMAGE CLASSIFICATION USING KERAS**" is an authentic record of my own work as requirements of Industrial Training during the period from **______1 May 2019____** to **_____8 August 2019_____** for the award of a degree of B.Tech (Computer Science), Manav Rachna University, Delhi NCR, India, under the guidance of Mr. Babloo Saha (Scientist 'D').

Deepika Ghotra

Student ID: 2K16CSUN01014

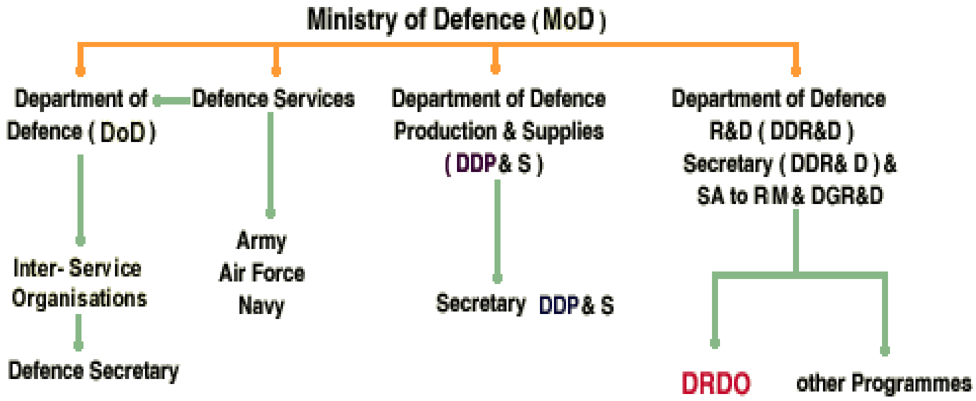
Date: ___1 August,2019________

INDEX

1. INTRODUCTION TO ORGANIZATION

1.1 ABOUT DRDO

DRDO functions as a wing of the Department of Defence Research and Development (DD R&D). Its hierarchical position is shown below:



DRDO was formed in 1958 from the amalgamation of the then already functioning Technical Development Establishment (TDEs) of the Indian Army and the Directorate of Technical Development & Production (DTDP) with the Defence Science Organization (DSO). DRDO was then a small organization with 10 establishments or laboratories. Over the years, it has grown multi-directionally in terms of the variety of subject disciplines, number of laboratories, achievements and stature.

Today, DRDO is a network of 52 laboratories which are deeply engaged in developing research technologies covering various disciplines, like aeronautics, armaments, electronics, combat vehicles, engineering systems, instrumentation, missiles, advanced computing and simulation, special materials, naval systems, life sciences, training, information systems and agriculture.

VISION

"Make India prosperous by establishing world class science and technology base and provide our Defence Services decisive edge by equipping them with internationally competitive systems and solutions."

MISSION

- Design, develop and lead to production state-of-the-art sensors, weapon systems, platforms and allied equipment for our Defence Services.
- Provide technological solutions to the Services to optimize combat effectiveness and to promote well-being of the troops.
- Develop infrastructure and committed quality manpower and build strong indigenous technology base.

DRDO LABS are deeply engaged in developing Defence technologies covering various fields, like aeronautics, armaments, electronic and computer sciences, human resource development, life sciences, materials, missiles, combat vehicles development and naval research and development. The organization includes more than 5,000 scientists and about 25,000 other scientific, technical and supporting personnel. Several major projects for the development of missiles, armaments, light combat aircrafts, radars, electronic warfare systems etc. are on hand and significant achievements have already been made in several such technologies.

DRDO provides technological solutions to the Services to optimize combat effectiveness and to promote well-being of the troops and lead to production state-of-the-art sensors, weapon systems, platforms and allied equipment for our Defence Services [1].

1.2 RESEARCH AREAS OF DRDO

DRDO dedicatedly working towards enhancing self-reliance in Defence Systems and undertakes design & development leading to production of world class weapon systems and equipment in accordance with the expressed needs and the qualitative requirements laid down by the three services.

DRDO is working in various areas of military technology which include aeronautics, armaments, combat vehicles, electronics, instrumentation engineering systems, missiles, materials, naval systems, advanced computing, simulation and life sciences. DRDO while striving to meet the Cutting edge weapons technology provides ample spinoff benefits to the society at large thereby contributing to the nation building.

1.3 THRUST AREAS OF DRDO

- ✔ Aeronautics
- ✔ Armaments
- ✔ Combat Engineering
- ✔ Electronics
- ✔ Life Science
- ✔ Materials
- ✔ Missiles
- ✔ Naval System

1.4 ABOUT ISSA

The idea to apply Operational Research (OR) and Systems Analysis techniques in the Indian Defence environment was conceived as early as the year 1959. A small group called Weapon Systems Analysis Group (WEG) was created to carry out Operational Research and Cost-effectiveness studies for weapons and equipments for the three Services, inter-service Organizations and the Ministry of Defence.

With the diversification of activities, WEG was renamed as Scientific Evaluation Group (SEG) in the year 1963 and later in the year 1968, SEG became a fully fledged directorate named as Directorate of Scientific Evaluation (DSE). The directorate continued to provide consultancy services to Services HQrs, SA to RM, DRDO HQrs and the Ministry of Defence in the fields of strategic Defence planning and analysis, weapons evaluation, damage assessment, performance evaluation and OR applications, etc. During this period, a large number of systems analysis studies were carried out. The results and recommendations of these studies contributed in the top-level decision making process.

In the year 2003, CASSA was merged with ISSA with an objective to synergies systems analysis activities and war gaming development processes under integrated combat environment. With this, ISSA has grown into a nodal systems analysis laboratory of DRDO, specializes in systems analysis, modeling & simulation of Defence systems using state-of-the-art info-technologies such as Computer Networking, Software Engineering, Distributed Database, Distributed Simulation, Web Technologies, Situational Awareness, and Soft-Computing techniques in development of complex simulation products.

1.4.1 AREAS OF WORK

ISSA has been identified as a nodal systems analysis laboratory in DRDO. It identifies the user needs and conducts system studies on the current and futuristic warfare requirements and undertakes development of integrated computer simulation models and software test beds for evolution and evaluation of strategies and tactics. This software serve as tools for analyses, using which ISSA provided turnkey solutions to the user’s problems.

Major areas of activities are:

- **Development of Computer War game Software for Land, Naval and Air Warfare Sensors and Weapon Systems Modeling**
    - Radar, EW, IW systems modeling and software development

- Decision support
- Combat modeling and simulation

- **Battlefield Damage Assessment**
  - Vulnerability studies
  - Weapon performance modeling

- **Network Centric Warfare**
  - Shared situation awareness algorithms
  - Information age combat models
  - Network centric integrated air Defence concepts

- **Strategic Systems Studies and Analysis**
  - Force mix studies
  - Strategic missile air Defence
  - Threat assessment and strategic planning

- **Operational Analysis**
  - Tactics development
  - Mission planning

- **Planning & Training Systems- Tactical & Strategic Decision Support**
  - Air Defence and electronic warfare simulation
  - Operational information dissemination

- **Reliability Analysis**
  - Life Cycle Cost Analysis

2. PROJECT OBJECTIVE

The main objective of this project is to create an image classification model that can classify various images into categories. We will be classifying various images related to military airplanes like Chinook ,Mikoyan MiG-21 ,Boeing 737 etc. from Air Force Bases, with images of bridges, buildings, airbases, etc. This model will help the war aircraft identify various military-related structures and will also help in identifying the enemy's military bases. For this purpose, we will be using Deep Learning. The model that we will create will be a Convolutional Neural Network(CNN) using Keras libraries.

The intent of the classification process is to categorize all pixels in a digital image into one of several land cover classes, or "themes.". This categorized data may then be used to produce thematic maps of the land cover present in the image. Normally, multispectral data is used to perform the classification. The objective of image classification is to identify and portray, as a unique gray level (or color), the features occurring in an image in terms of the object or type of land cover these features actually represent on the ground.

3. Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems with 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 algorithms are often categorized as supervised or unsupervised.

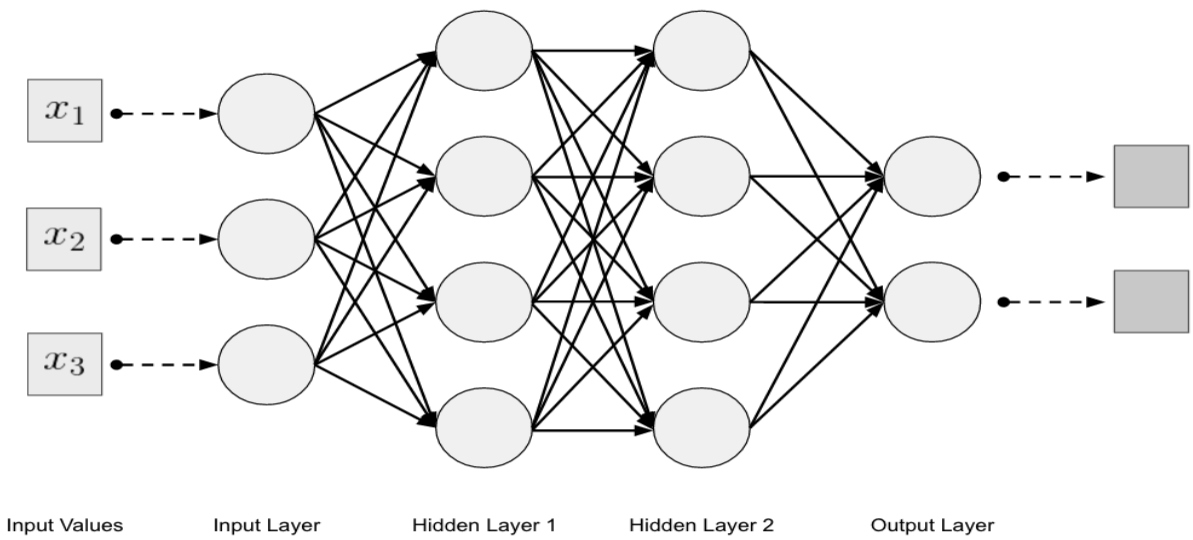
- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
- In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

4. Deep Learning

Deep 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 those of human experts.

4.1 Basic Architecture of Neural Networks



The basic architecture of a Neural Network consists of an input layer, hidden layers and output layer.

The input layer consists of the input values or the independent variables. All the inputs are fed in the model through this layer.

A hidden layer in neural network is a layer in between input layer and output layer, where artificial neurons take in a set of weighted inputs and produce an output through an activation function. The number of hidden layers depends on how deep is your model or architecture.

The output layer consists of the dependent variables. The data after processing is made available at the output layer.

5. Convolutional Neural Network

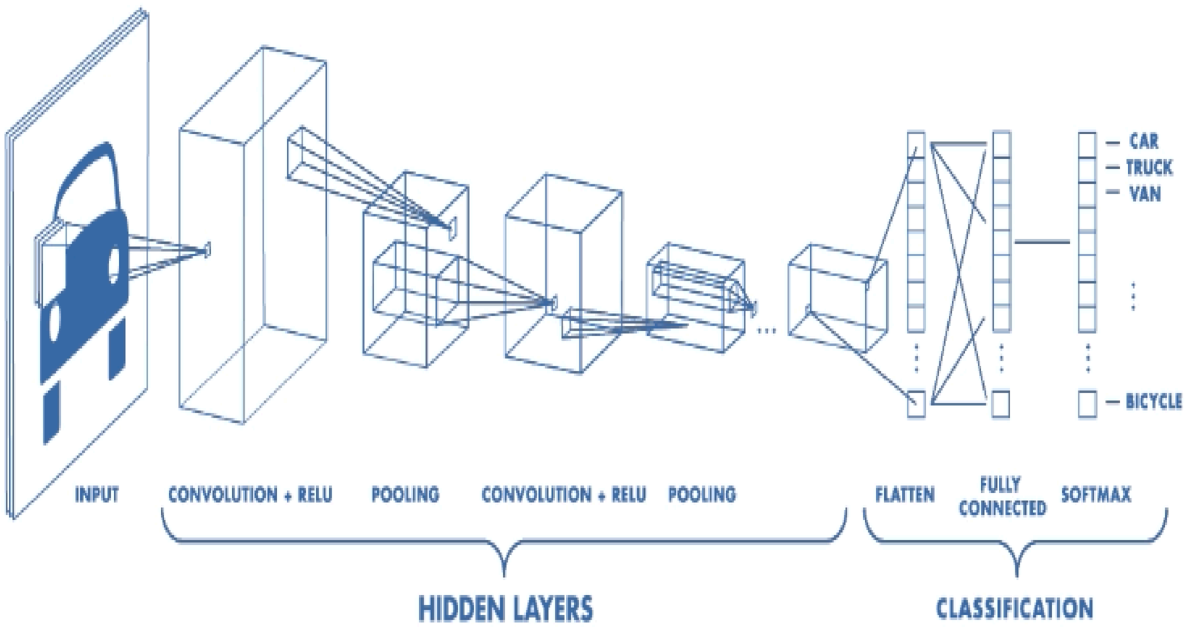
In neural networks, Convolutional neural network (CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg. Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution.

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters, Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

Since our project requires image classification, we will be using Convolutional Neural Network(CNN) using Keras.

5.1 Architecture of Convolutional Neural Network



5.1.1 Convolution Layer:

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter. Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying filters.

5.1.2 Non Linearity (ReLU):

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = \max(0,x)$. ReLU's purpose is to introduce non-linearity in our CNN. Since, the real world data would want our CNN to learn would be non-negative linear values.

There are other non linear functions such as tanh or sigmoid can also be used instead of ReLU. ReLU is widely used since performance wise it is better than other two.

5.1.3 Pooling Layer:

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

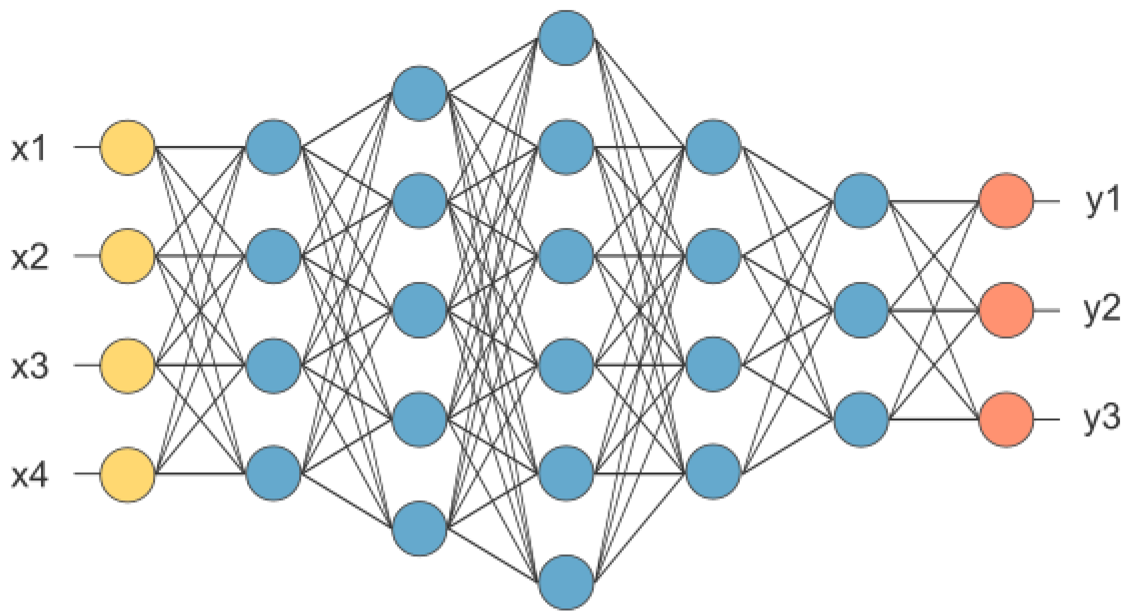
Max pooling take the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

5.1.4 Flattening:

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer. In other words, we put all the pixel data in one line and make connections with the final layer.

5.1.5 Fully Connected Layer:

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network.



In the above diagram, the feature map matrix will be converted as a vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs.

6. Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling.

6.1 Steps for setting the environment

1. Download and Install Anaconda from https://www.anaconda.com/distribution/

Installing Theano

2. pip install --upgrade --no-deps git+git://github.com/Theano/Theano.git

Installing Tensorflow

Install Tensorflow from the website:
3.https://www.tensorflow.org/versions/r0.12/get_started/os_setup.html

Installing Keras

4. pip install --upgrade keras

7. Source Code

Convolutional Neural Network

Part 1 - Building the CNN

Importing the Keras libraries and packages

```
from keras.models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.layers import Dense
```

 # Initialising the CNN

```
classifier = Sequential()
```

 # Step 1 - Convolution

```
classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
```

 # Step 2 - Pooling

```
classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

 # Adding a second convolutional layer

classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))

classifier.add(MaxPooling2D(pool_size = (2, 2)))

 # Step 3 - Flattening

classifier.add(Flatten())

 # Step 4 - Full connection

classifier.add(Dense(output_dim = 128, activation = 'relu'))

classifier.add(Dense(output_dim = 1, activation = 'sigmoid'))

 # Compiling the CNN

classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

 # Part 2 - Fitting the CNN to the images

from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,

shear_range = 0.2,

zoom_range = 0.2,

horizontal_flip = True)

```
test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('Dataset/Training',
                                                 target_size = (64, 64),
                                                 batch_size = 8,
                                                 class_mode = 'binary')

test_set = test_datagen.flow_from_directory('Dataset/Test',
                                            target_size = (64, 64),
                                            batch_size = 8,
                                            class_mode = 'binary')

classifier.fit_generator(training_set,
                         samples_per_epoch = 135,
                         nb_epoch = 30,
                         validation_data = test_set,
                         nb_val_samples = 28)
```

 # Part 3 - Making new Predictions

```
 import numpy as np

 from keras.preprocessing import image

 import glob


 prediction = []

 filePaths = glob.glob("Dataset/single_prediction/*.JPG")


for file in filePaths:

        test_image = image.load_img(file,

                target_size = (64, 64))

        test_image = image.img_to_array(test_image)

        test_image = np.expand_dims(test_image, axis = 0)

        result = classifier.predict(test_image)

        training_set.class_indices


        if result[0][0] == 1:

                p = 'Airplane’'

        if result[0][0] == 0:

                p = 'Unrecognized'
```

prediction.append(p)

8. Output

```
Epoch 27/30
16/16 [==============================] - 3s 202ms/step - loss: 0.0709 - acc: 0.9766
- val_loss: 0.6532 - val_acc: 0.8214
Epoch 28/30
16/16 [==============================] - 3s 204ms/step - loss: 0.0252 - acc: 1.0000
- val_loss: 0.8188 - val_acc: 0.8929
Epoch 29/30
16/16 [==============================] - 3s 209ms/step - loss: 0.0249 - acc: 0.9922
- val_loss: 0.9282 - val_acc: 0.8214
Epoch 30/30
16/16 [==============================] - 3s 204ms/step - loss: 0.0454 - acc: 0.9922
- val_loss: 0.7899 - val_acc: 0.8929
Out[15]: <keras.callbacks.History at 0x2425c6aacc0>
```

9. Sample Images

10. Conclusion and Future Work

In this report, we precisely explained our project and showed some background material. After that, we introduced our solution in detail. To achieve better performance, we implemented two convolutional layers so that our model learns better.

In the future, we will explore more to achieve better performance. For instance, we will try to change the architecture and parameter settings of the Deep Neural Network based on the feature visualization of different layers' feature maps.

Finally, I would like to conclude that in the six weeks while i was working on this project,
I learned many new technologies and concepts.
Thus, I was able to understand in greater detail the various software engineering processes and was able to apply them to my project.
With this enduring and simulating experience, I admit that the people of this organization has really enlightened me. With due respect, I want to express our heart-felt thanks to all for their support and cooperation towards the completion of our project.

11. References

Web Links

1. http://www.google.com
2. https://keras.io/
3. https://www.anaconda.com/distribution/
4. https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5
5. https://en.wikipedia.org/wiki/Convolutional_neural_network