MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE

JAN-MAY 2021

A REPORT ON INTERNSHIP AS TECHNICAL CONTENT DEVELOPER AT RAPIDQUEST SOLUTIONS

SUBJECT CODE - 160801

SUBMITTED TO: **SUBMITTED BY:**

PROF. VIKAS SEJWAR   **SAFAL KHAMPARIYA**

                                                            0901IT171045

INTERNSHIP CERTIFICATE

Date: June 03, 2021

To Whom It May Concern

This letter is to certify that **Mr. Safal Khamparia** has successfully completed his internship program at RapidQuest Solutions Pvt. Ltd. He was actively engaged during the term of **Feb 15, 2021** to **June 1, 2021**.

Safal worked with our content development and machine learning teams and was actively involved in the projects and tasks assigned to him. During the span, we found him to be a punctual and a hardworking person.

We wish him a bright future.

Best Regards,

Mthakkar

Digitally Signed By:
Mihir Thakkar
Date: 2021.06.03
10:20:11

Mihir Thakkar
Director and CEO

RapidQuest

Table of Contents

ACKNOWLEDGEMENT

Firstly I would like to thank **Founder & CEO** of **RapidQuest solutions Mr. Mihir Thakkar** for giving me this opportunity to do this internship within the organisation.

I have completed this internship under the guidance of **Mr. Mihir Thakkar** and without him this journey would have not been possible.

I would also like to thank the entire team of **Code Heroku** for their patience and assistance during my online internship.

I would like to thank the college administration of **Madhav Institute of Technology and Science** for giving me this opportunity to go for an internship during the final semester of my college.

I would also like to thank **Prof. Vikas Sejwar** my institute mentor for his constant support throughout this internship.

INTRODUCTION

Introduction of the company:

Established in 2018 RapidQuest specializes in providing AI solutions for software products. They are experts in building and deploying cutting-edge Deep Learning based AI solutions.

The organization that I interned with is called **Code Heroku** which is a subsidiary of RapidQuest solutions.

Code Heroku is an online education platform focused on delivering high quality computer science and engineering courses to students in developing countries. Code Heroku is building an ecosystem around students to nurture their creativity, innovation and most importantly their desire to learn more.

Information about the position:

As an intern with Rapidquest solutions I worked as a Technical content developer and was a part of the machine learning and content development team.

Tasks & Responsibilities:

- Creating new programming tutorials videos and courses.
- Answering questions from students on website forums.
- Managing course Github repositories

PROJECTS UNDERTAKEN

Building a web app for object detection using YOLO object detection script:

Software Requirement Specification :

- Programming Language : Python
- Libraries and frameworks : OpenCV, numpy, flask, json
- IDE : Jupyter Notebook

Object Detection :

Object detection is basically a computer vision technique that allows us to identify and locate different kind of objects in an image or a video.Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in (or how they move through) a given scene.

YOLO object detection script :

Yolo or "you only look once" is the state of the art object detection model. Yolo applies a single neural network to the full image.This network then divides the image into regions and predicts bounding boxes and probabilities for each region.These bounding boxes are weighted by the predicted probabilities.

STRUCTURE OF YOLO :
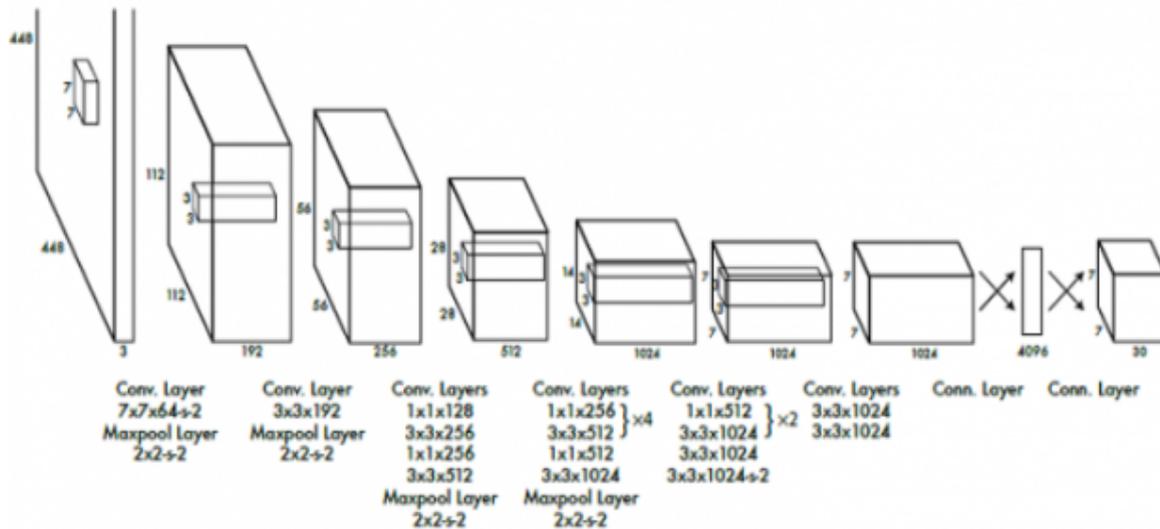
The detection network has 24 Convolution Layers and 2 fully connected layers.Alternating 1X1 convolution layers reduce the feature space from preceding layers.

FLASK :

Flask is a lightweight web-server gateway interface.It is basically a specification that describes how a web server communicates with web applications and how web applications can be chained together to process one request.

In this project flask was used to deploy model as a web app.

The tasks that I undertook in this project were:

- Created a python program using YOLO object detection script, numpy and OpenCV DNN.

- The program had a function that took an image as an input and returned the image with bounding boxes around the located objects.
- Created an index.html file which had a form element that can take an image as an input.
- Created an app.py file that used flask to link the html file with python program and deployed the Object detector as a web app.
- Created a video tutorial that explained each and every aspect of this project including the hands on part.

Code snippets :


```python
from flask import Flask, request, jsonify, Response, render_template
import cv2
import numpy as np
import json

def object_detection_api(image):

    confidenceThreshold = 0.5
    NMSThreshold = 0.3

    modelConfiguration = 'cfg/yolov3.cfg'
    modelWeights = 'yolov3.weights'

    labelsPath = 'coco.names'
    labels = open(labelsPath).read().strip().split('\n')

    np.random.seed(10)
    COLORS = np.random.randint(0, 255, size=(len(labels), 3), dtype="uint8")

    net = cv2.dnn.readNetFromDarknet(modelConfiguration, modelWeights)

    (H, W) = image.shape[:2]

    # Determine output layer names
    layerName = net.getLayerNames()
    layerName = [layerName[i[0] - 1] for i in net.getUnconnectedOutLayers()]

    blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416), swapRB=True, crop=False)
    net.setInput(blob)
    layersOutputs = net.forward(layerName)

    boxes = []
    confidences = []
```

```python
@app.route('/object_detector',methods=['GET','POST'])

def main():

    if request.method=='GET':
        return render_template('index.html')

    if request.method=='POST':
        file=request.files['image']
        npimg=np.fromfile(file,np.uint8)
        img=cv2.imdecode(npimg,cv2.IMREAD_COLOR)


        # image_path=request.args.get('image')
        # img=cv2.imread(image_path)
        result=object_detection_api(img)

        _, img_encoded=cv2.imencode('.png', result)
        response=img_encoded.tostring()

        return Response(response=response, status=200, mimetype='image/png')
        # return jsonify(result)


if __name__=='__main__':
    app.run()
```

Results achieved :

Input image

Output image after object detection:
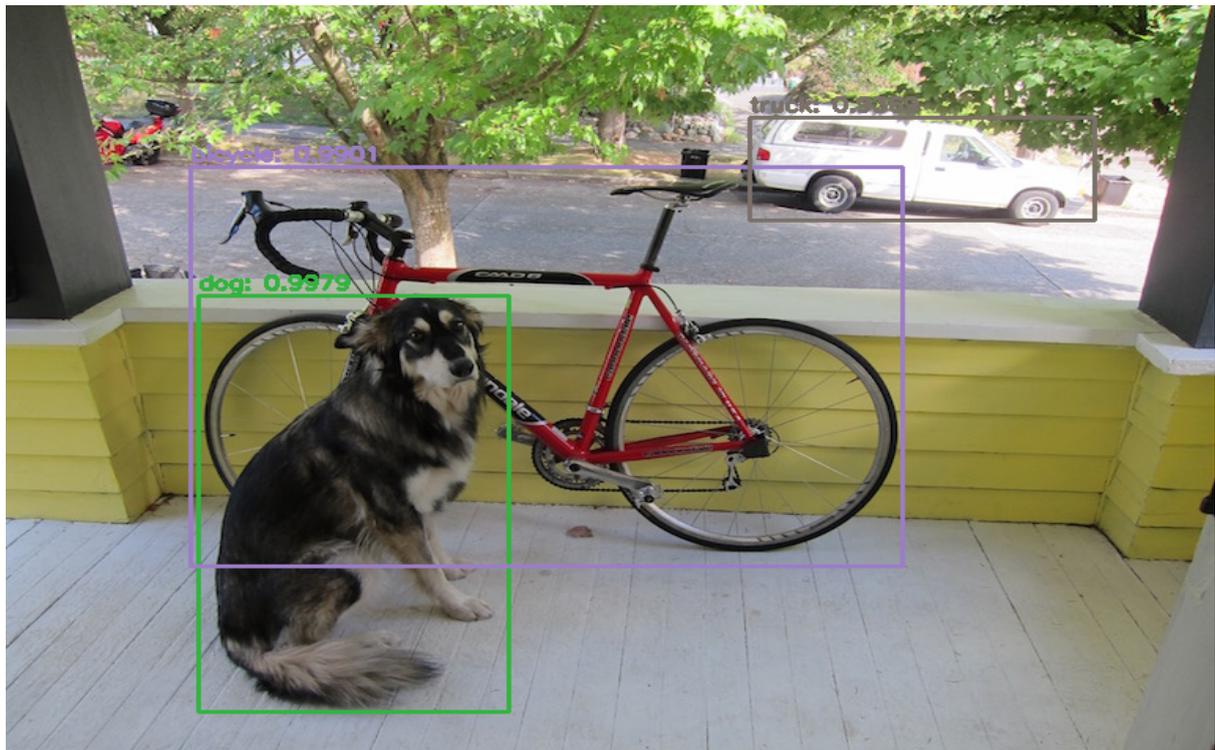
Sentiment analysis of text data using Tensorflow :

Software Requirement Specification :

Programming language : Python

API's : Tensorflow, keras

Libraries and Frameworks : Pandas, Numpy

IDE : Jupyter Notebook

Sentiment analysis :

Sentiment analysis (opinion mining) is a text mining technique that uses machine learning and natural language processing (nlp) to automatically analyze text for the sentiment of the writer (positive, negative, neutral, and beyond).

The overall purpose of text mining is to derive high-quality information and actionable insights from text, allowing businesses to make informed decisions.

Powerful machine learning algorithms can easily recognize statements as *Positive, Negative, or Neutral*. And you can get even more granular results when you put aspect-based sentiment analysis into practice.

Applications of sentiment analysis :

- This technique is widely used for understanding social sentiments for trending topics by gathering data from social media websites like twitter and facebook.
- It is also used by organizations and companies to understand the overall public sentiments of their products by gathering user reviews.
- It can also be used to find wether a post or a tweet should be removed or not and wether it is within the guidelines or not.
- Stock trading companies also use sentiment analysis to

find companies that show a positive sentiment in news articles.

Word Embeddings :

Word embeddings give us a way to use an efficient dense representation in which similar words have similar encoding.An embedding is a dense vector of floating point values (we specify the length of the vector).The values for the embeddings are trainable parameters.

Dataset used and objective :

Dataset used was the imdb dataset which contains user reviews for various movies and their corresponding sentiments which are either positive or negative.

The objective of this exercise was to create and train a model which can classify the sentiments of reviews wether it is positive or negative.

Structure of the neural network model :

- Our model is a neural network which consists of 4 layers.
- The first layer is a word embedding layer which converts the integer encoded reviews into embeddings.
- The second layer is global average pooling layer which basically takes the average value of embeddings and creates an vector that contains average of values.
- The third layer of the model is a dense layer which

contains 16 units and uses 'relu' as an activation function.

- Fourth and the final layer is the output layer which has one unit and uses 'sigmoid' activation function.

The tasks that I undertook during this projects were :

- Preprocessing of dataset.
- The dataset consisted of text data the strings were tokenized to get an array of strings.

- Integer encoded this array using a vocabulary list.

- Converting the integer encoded array into word embeddings.

- Creating a neural network model to classify the sentiments of the reviews taking word embeddings as an input.

- Creating PPT's and video tutorial explaining each and every aspect of the project.

Code snippets :

Sentiment analysis ☆

File Edit View Insert Runtime Tools Help · Last edited on February 26

💬 Comment · 👥 Share · ⚙ ·

+ Code · + Text · Connect ▾ · ✎ Editing · ▲

```python
import tensorflow as tf
import tensorflow.keras as keras
import pandas as pd
import numpy as np
```

```python
imdb_reviews=pd.read_csv("/content/imdb_reviews.csv")
test_reviews=pd.read_csv("/content/test_reviews.csv")
```

```python
imdb_reviews.head()
```

	Reviews	Sentiment
0	<START this film was just brilliant casting lo...	positive
1	<START big hair big boobs bad music and a gian...	negative
2	<START this has to be one of the worst films o...	negative
3	<START the <UNK> <UNK> at storytelling the tra...	positive
4	<START worst mistake of my life br br i picked...	negative

```python
word_index=pd.read_csv("/content/word_indexes.csv")
```

```python
word_index.head(n=10)
```

Words Indexes

Sentiment analysis ☆

File Edit View Insert Runtime Tools Help · Last edited on February 26

💬 Comment · 👥 Share · ⚙ ·

+ Code · + Text · Connect ▾ · ✎ Editing · ▲

```python
def encode_sentiments(sentiment):
    if sentiment=='positive':
        return 1
    else:
        return 0

train_labels=train_labels.apply(encode_sentiments)
test_labels=test_labels.apply(encode_sentiments)
```

```python
train_data=keras.preprocessing.sequence.pad_sequences(train_data,value=word_index["<PAD>"],padding='post',maxlen=500)
test_data=keras.preprocessing.sequence.pad_sequences(test_data,value=word_index["<PAD>"],padding='post',maxlen=500)
```

```python
model=keras.Sequential([keras.layers.Embedding(10000,16,input_length=500),
                        keras.layers.GlobalAveragePooling1D(),
                        keras.layers.Dense(16,activation='relu'),
                        keras.layers.Dense(1,activation='sigmoid')])
```

```python
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

```python
history=model.fit(train_data,train_labels,epochs=30,batch_size=512,validation_data=(test_data,test_labels))
```

```
Epoch 1/30
49/49 [==============================] - 3s 44ms/step - loss: 0.6926 - accuracy: 0.5385 - val_loss: 0.6893 - val_accuracy: 0.6035
Epoch 2/30
49/49 [==============================] - 2s 31ms/step - loss: 0.6862 - accuracy: 0.6723 - val_loss: 0.6764 - val_accuracy: 0.7321
Epoch 3/30
49/49 [==============================] - 1s 30ms/step - loss: 0.6683 - accuracy: 0.7517 - val_loss: 0.6476 - val_accuracy: 0.7423
```

Deep Learning Trends in 2021 :

In today's world everyday there is some new advancement in technology.So in this project I created a tutorial where I tried to explain four new trend's in the field of deep learning which were

- Improved natural language processing
- AI become's better at gameplay
- Deepfakes
- Efficient neural networks for embedded devices

Improved natural language processing :

Although natural language processing has it's roots back in 1950's one of the recent breakthrough's was achieved recently when in May 2020 OpenAI released GPT3.

GPT 3 :

Generative pretrained transformers or GPT 3 is an autoregressive language model which can produce human like text. It can write stories, scripts, articles, poems and it can even write code.

It is also the largest built neural network till date as it has 175 billion parameters and it is trained over 500 GB of text data.

GPT 3 is a model that take's an input which is a sequence of words and predict's the next word in the sequence.

Accuracy of GPT 3 :
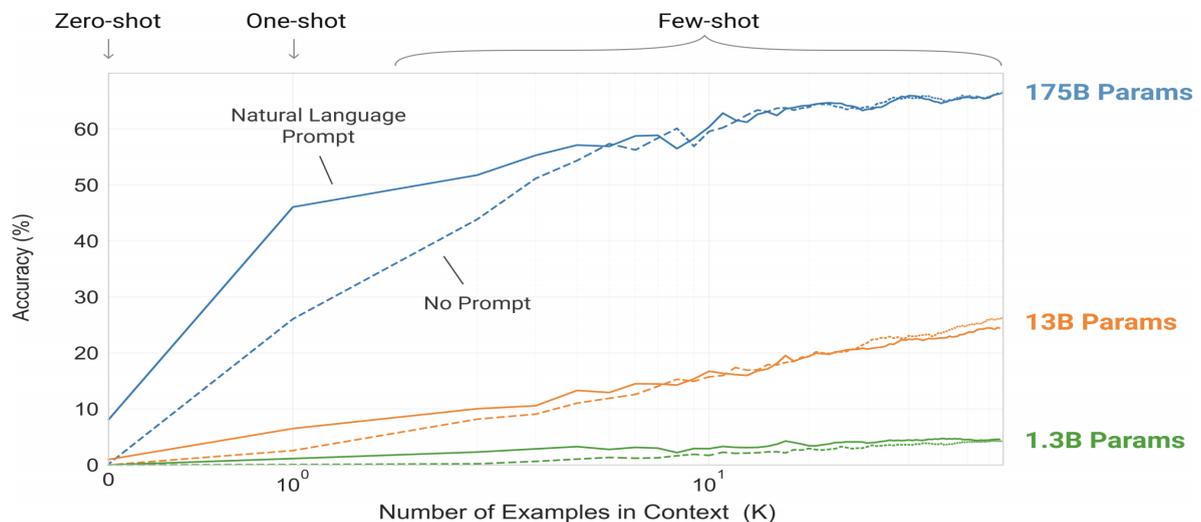
AI become's better at gameplay :

The long term goal of AI is to solve advanced real world challenges & games have served as a stepping stone in this path for decades from Backgammon in 1992 to chess in 1997 to Atari in 2013.

The major breakthrough was achieved on April 13th 2019 when OpenAI 5 became the first AI system to defeat the world champion's at an Esports game.

Open AI 5 is a model created by OpenAI that has been trained to play Dota 2.

Reason's for this being a major breakthrough :
- Long time horizons
- Partial observability
- High dimensionality

- Complex rules
- Long term strategic planning

OpenAI used a technique called reinforcement learning to train the OpenAI 5 model.

Structure of OpenAI 5 :
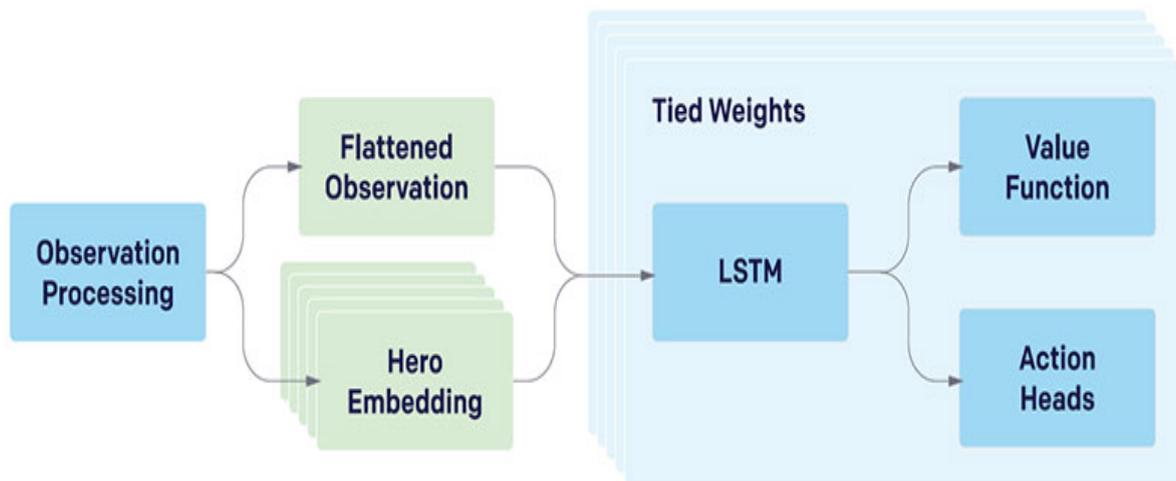
Deepfakes :

Deepfakes are synthetic media[2] in which a person in an existing image or video is replaced with someone else's likeness.

While the act of faking content is not new, deepfakes leverage powerful techniques from machine learning and artificial intelligence to manipulate or generate visual and audio content with a high potential to deceive.

The main machine learning methods used to create deepfakes are based on deep learning and involve training

generative neural network architectures, such as autoencoders or generative adversarial networks (GANs).

Deepfakes have garnered widespread attention for their uses in celebrity pornographic videos, revenge porn, fake news, hoaxes, and financial fraud.This has elicited responses from both industry and government to detect and limit their use.

Efficient neural networks for embedded devices :

State of the art neural network's are expensive. These expensive model's cannot be processed in device's like mobiles or other devices which have low processing power.

So the researcher's at google came up with a model called Efficient net which along with increasing the accuracy of the model tries to keep the computations less so that efficiency can be achieved and model can be processed in embedded devices.

Efficient net :

When we want to scale up our model for better accuracy we can do three things

- Increase the number of layers.
- Increase the number of units per layer.
- Increase the resolution of the image.

The researchers came to a conclusion that we can scale up the models while maintaining the efficiency by compound scaling i.e increase all dimensions in fixed ratio.
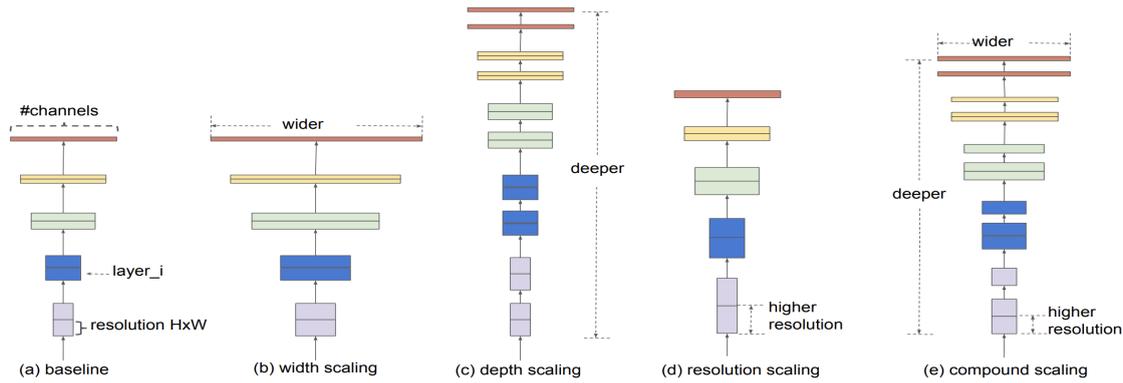
Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

The tasks that I undertook during this project :

- Researching about recent trends and major breakthroughs in the field of deep learning.
- Selecting four major trends.
- Creating a video tutorial explaining each trend and there future prospects in detail.

OpenCV installation for C++ using MinGW compiler and CMake :

Whenever someone wants to create a project in the field of computer vision mostly he/she is told to use python as the programming language so I created a tutorial to teach people how we can use OpenCV in C++ without any ide and create computer vision projects.

Software requirement specification :

Programming Language : C++

Compiler : MinGW-w64

Libraries and frameworks : OpenCV

Build tool : CMake

OpenCV :

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.The library has more than 2500 optimized algorithms which includes a set of both classic and state-of-the-art computer vision and machine learning algorithms.

CMake :

In software development CMake basically is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of our choice.

The tasks that I undertook during this project were :

- Researching and finding out a way to create OpenCV projects in C++ without using any IDE.
- Creating a basic OpenCV program in C++ in a normal text editor.
- Creating CMakeLists.txt file to link OpenCV libraries for MinGW compiler with MinGW compiler.

Code snippets :
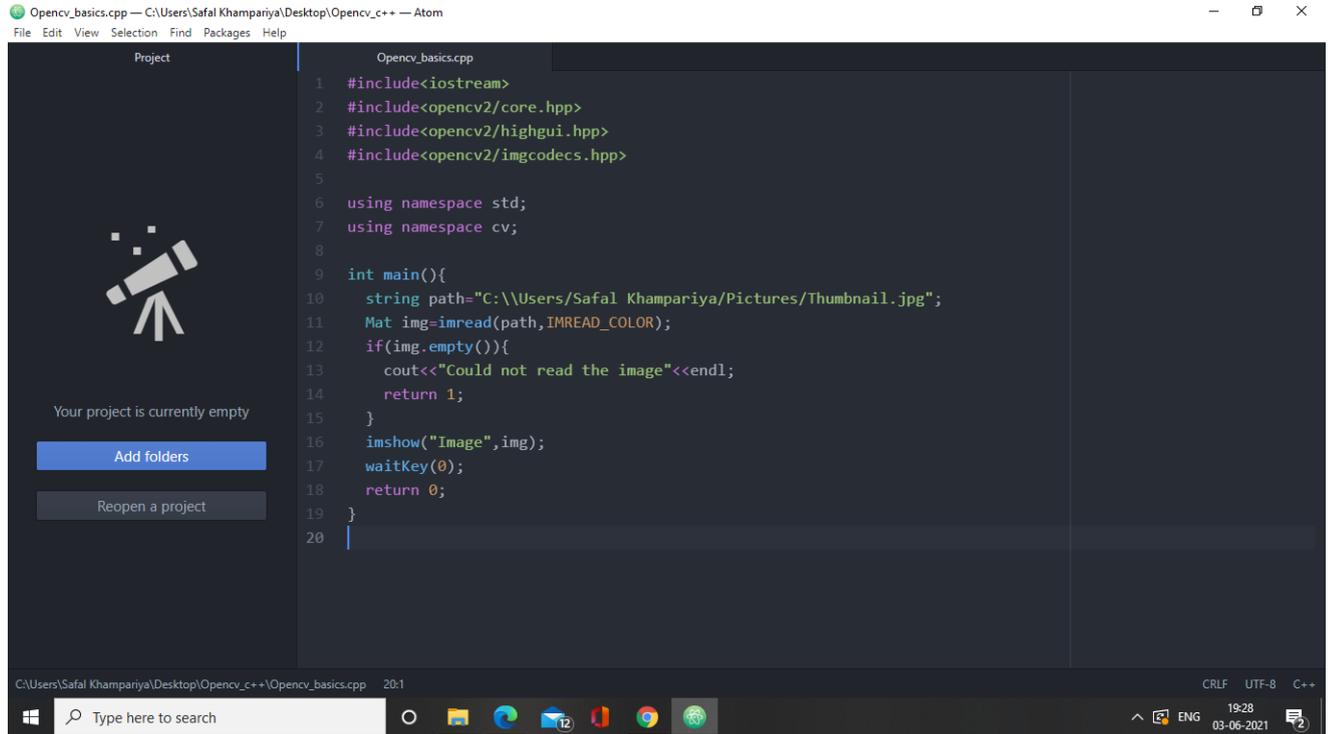
Opencv_basics.cpp — C:\Users\Safal Khampariya\Desktop\Opencv_c++ — Atom

File   Edit   View   Selection   Find   Packages   Help

```cpp
#include<iostream>
#include<opencv2/core.hpp>
#include<opencv2/highgui.hpp>
#include<opencv2/imgcodecs.hpp>

using namespace std;
using namespace cv;

int main(){
    string path="C:\\Users/Safal Khampariya/Pictures/Thumbnail.jpg";
    Mat img=imread(path,IMREAD_COLOR);
    if(img.empty()){
        cout<<"Could not read the image"<<endl;
        return 1;
    }
    imshow("Image",img);
    waitKey(0);
    return 0;
}
```

Project : Your project is currently empty

Add folders

Reopen a project

C:\Users\Safal Khampariya\Desktop\Opencv_c++\Opencv_basics.cpp   20:1   CRLF   UTF-8   C++
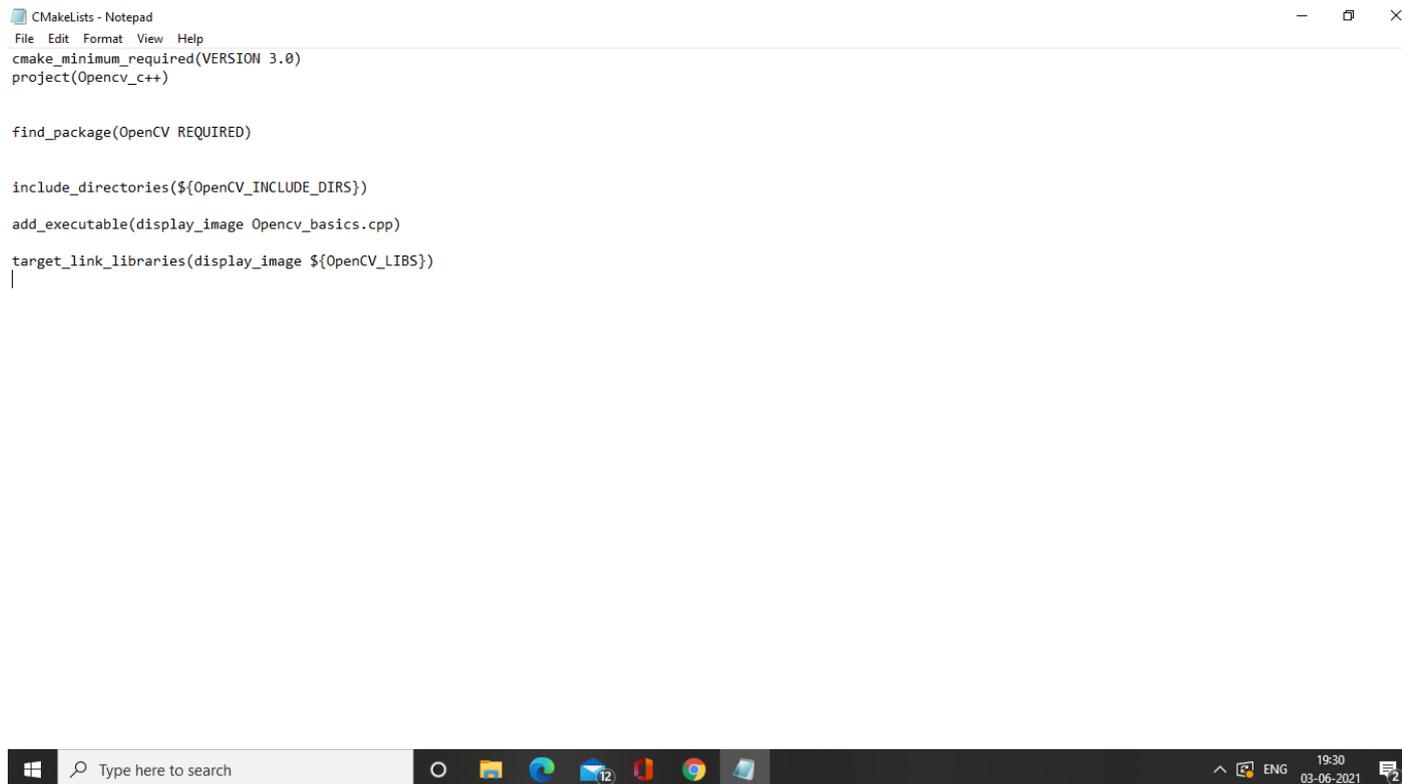
CMakeLists - Notepad

File   Edit   Format   View   Help

```cmake
cmake_minimum_required(VERSION 3.0)
project(Opencv_c++)

find_package(OpenCV REQUIRED)

include_directories(${OpenCV_INCLUDE_DIRS})

add_executable(display_image Opencv_basics.cpp)

target_link_libraries(display_image ${OpenCV_LIBS})
```

AWS for Machine Learning :

Cloud computing :

Cloud computing is the on-demand availability of computer system resources, especially data storage (cloud storage) and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet. Large clouds, predominant today, often have functions distributed over multiple locations from central servers. If the connection to the user is relatively close, it may be designated an edge server.
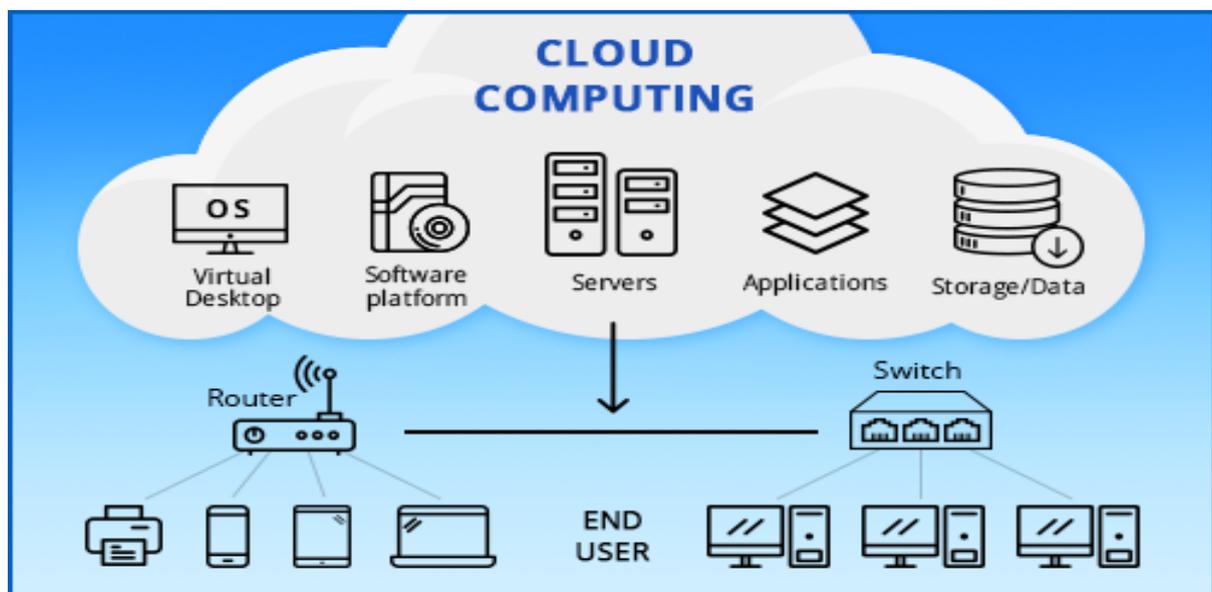
Amazon web services(AWS) :

AWS or amazon web services is a subsidiary of Amazon that purchases and maintains IT resources such as servers, storage

etc. and provides these resources to individuals, companies or governments on a metered pay as you go basis.

Virtual Machines :

Virtual machine is no different than the physical computer or server and it can do everything that your computer does.The only difference is that your computer has hardware the virtual machines are software defined computers within physical servers.

AWS EC2 :

- AWS Elastic compute cloud or EC2 is a web service that provides secure and resizable computing capacity in the cloud.It gives you total control of your computing environment.
- With amazon EC2 you can launch as many virtual machines as you want and run them on Amazon's computing environment.

The tasks that I undertook in this project were :

- Creating an AWS Ec2 instance with required configurations so that the server can be accessed remotely.
- Connecting to the server through command line using SSH.
- Installing and using Jupyter notebook in the server.
- Accessing the jupyter notebook running in server through the local browser.
- Creating PPT's and video tutorial explaining each of the steps.
- Creating a blog explaining the steps on companies website.

Code snippets :


```
GNU nano 4.8                    /home/ubuntu/.bashrc                    Modified
#Jupyter path
export PATH="/home/ubuntu/anaconda3/bin:$PATH"

# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line
```



```
ubuntu@ip-172-31-20-196:~$ which python
ubuntu@ip-172-31-20-196:~$ nano ~/.bashrc
ubuntu@ip-172-31-20-196:~$ source .bashrc
(base) ubuntu@ip-172-31-20-196:~$ which python
/home/ubuntu/anaconda3/bin/python
(base) ubuntu@ip-172-31-20-196:~$ ipython
Python 3.8.5 (default, Sep  4 2020, 07:30:14)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.19.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from IPython.lib import passwd

In [2]: passwd()
Enter password:
Verify password:
Out[2]: 'sha1:769a9ac85257:7223fe35a2786b71d65fb63804e1729038475333'

In [3]: exit
(base) ubuntu@ip-172-31-20-196:~$
```

CONCLUSION

From my internship at RapidQuest solutions I learnt a lot.I was able to get a better understanding of how to work in a professional environment and become a professional individual.I was also able to get knowledge of different topics related machine learning and data science while I was creating tutorials on them.I also got a better understanding of how the rapidly growing Edu-Tech and Elearning sector works and how it is also creating a positive impact on the quality of education.I enjoyed working with the professional and talented team of one of India's fastest growing startups and getting guidance about the new technologies that are changing this world was a life changing experience for me.However, since machine learning and artificial intelligence are very vast fields of study I still have a long way ahead in understanding and getting more insights in the different aspects of these domains.

Overall I found this internship very intresting and positive and I'm sure that I will be able to use the skills that I gained in this internship and in my life and career ahead.