

# **CELEBRITY IMAGE CLASSIFIER**

Internship Report submitted in partial fulfillment of the requirement for the award of the  
degree of

**BACHELOR OF TECHNOLOGY**  
**IN**  
**COMPUTER SCIENCE AND ENGINEERING**

By

**A.Vamshi Krishna**  
**20R11A0562**



**Department of Computer Science and Engineering**  
**Accredited by NBA**

**Geethanjali College of Engineering and Technology**  
**(UGC Autonomous)**

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi)  
Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

**August-2022**

# Geethanjali College of Engineering & Technology

(UGC Autonomous)

(Affiliated to JNTUH, Approved by AICTE, New Delhi)

Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA



## CERTIFICATE

This is to certify that the Internship Report entitled “**Celebrity Image Classifier**” is a bonafide work done by **A.Vamshi Krishna (20R11A0562)** in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in “**Computer Science and Engineering**” from Jawaharlal Nehru Technological University, Hyderabad during the year 2021-2022.

**HOD - CSE**

**Dr. A Sree Lakshmi**

**Professor**

Examiner

Name:

Designation:

## Internship Certificate:



# **Geethanjali College of Engineering & Technology**

**(UGC Autonomous)**

(Affiliated to JNTUH Approved by AICTE, New Delhi)

Cheeryal (V), Keesara(M), Medchal Dist.-501 301.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Accredited by NBA**



## **DECLARATION BY THE CANDIDATE**

I, **A.Vamshi Krishna** bearing **Roll No 20R11A0562** hereby declare that the Internship Report entitled “**Celebrity Image Classification**” is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**.

This is a record of bonafide work carried out by me in **1Stop** and the results embodied in this internship have not been reproduced or copied from any source. The results embodied in this Internship Report have not been submitted to any other University or Institute for the award of any other degree or diploma.

**A.Vamshi Krishna**  
**20R11A0562**

**Department of CSE,**

**Geethanjali College of Engineering and  
Technology, Cheeryal.**

## **ACKNOWLEDGEMENT**

I would like to thank **1Stop** for giving me the opportunity to do an internship within the organization. I also would like to thank all the people who worked along with me with their patience and openness. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Principal, Dr. Udaya Kumar Susarla , for the facilities provided to accomplish this internship. I would like to thank Head of the Department, Dr. A. Sree Lakshmi, for her constructive criticism throughout my internship.

## Introduction about Internship Organization

1stop is a strong platform with immense potential for students from anywhere and anytime to access cutting-edge Mentorship and Certification Program widespread with a larger number of domains and niches to explore learning. E-learning has been the fastest growing way for an aspirant to start their journey into pursuing their Dream Career.

1Stop brings academia & industry very close for a common goal of talent creation through experiential learning & development environment. The platform enables students to move in a successful career path to reach the expectations of industry. They are building the Next-Gen Talent pool with skills in emerging technologies i.e., Artificial Intelligence, Data Science, Internet of Things(IoT), Robotics, Blockchain, Quantum Computing and Cyber Security. The unique method of 1Stop is project-based learning, micro-skilling and Internships helps students in building their competency & get ready for industry. They bring students, educators and employers on a common platform to fill the gap between academia & industry.

**Methodology:** Collaborative learning in a project environment with industry interaction & mentoring.

**Learn:** Project based learning environment to understand the practical application of theoretical concepts by deeply engaging the students.

**Practice:** Practice through problem solving and project building can make a student an expert in his field of interest.

**Intern:** Collaborative working environment with industry mentorship to make young talent ready for real world challenges.

## **Main objectives of 1Stop**

- Well directed career guidance programs for educational institutions.
- Appropriate certification courses that suit the industry need.
- Train the trainers; expanded awareness about the current industry standards.
- Liaise with corporates to offer niche internships.

One formula they follow for success is 70:20:10 Rule:

- 10% Learn It Yourself: They got the largest library of in demand technologies with the best materials for you to choose when & if you need.
- 20% You Share, You Grow: The platform is designed to skill you up while you work with teams and experts as there are mentors from around the world.
- 70 % Say Hello to experience: There is nothing better than Project based learning and they provide the best of the real-life working experience to you from the comfort practicing anywhere.

## Training Schedule

Module	MODULE TITLE	TOPICS	TIME ALLOCATED	ASSESSMENT
Module 1	<b>AI_E-Learning Content</b>	Introduction to artificial intelligence, Introduction to python, Introduction to python programming, basics of python <u>programming</u> , introduction to matplotlib, introduction to machine learning un Supervised Learning	3 Hours	Assessment 1
Module 2	<b>AI - Hand Written Digit Classification with CNN from Personify Project - 1</b>	Introduction to tensor flow, introduction to keras, clustering, introduction to deep learning, natural language process , digital classification with CNN	1 Hours	Assessment 1

Module 3	<b>AI - News Classification using Natural Language Processing from Personify Project - 2</b>	News classification-1, News classification-2, News classification-3	2 Hours	Assessment 1
Module 4	<b>Recordings of Project Live Classes</b>	Hand written digit classification, News classification	2 Hours	Assessment 2
Module 5	<b>Celebrity Image Classifier - Internship Project from Personify</b>	Celebrity image classifier	2 Hours	Assessment 2



## TABLE OF CONTENTS

<b>S. No</b>	<b>Contents</b>	<b>Page No</b>
	<b>Abstract</b>	<b>01</b>
	<b>List of Figures</b>	<b>05</b>
	<b>List of Screenshots</b>	<b>08</b>
<b>1</b>	<b>Introduction</b>	<b>02</b>
<b>2</b>	<b>Work done/ Observation/ Duties performed</b>	<b>03</b>
<b>3</b>	<b>Related work/ Technologies used</b>	<b>10</b>
<b>4</b>	<b>Learning after Internship</b>	<b>10</b>
<b>5</b>	<b>Summary / Conclusion</b>	<b>11</b>
<b>6</b>	<b>Bibliography</b>	<b>12</b>

## **Abstract**

In our research, the problem is to find out a specific individual from a large image dataset. Our focus is on individuals' faces and their patterns. Faces from all over the world share many characteristics. However, there may be some differences, which we must investigate. Our task is to locate the appropriate individual and provide accurate information about him or her. To work with this challenge, we have selected Amazon celebrity Recognition. In the last few years, Deep Learning has been a very popular research topic in computer vision. It is a branch of Artificial Intelligence and a sub-field of Machine Learning [1]. It can achieve very significant results on a variety of image and pattern recognition problems, like image recognition, voice recognition, etc.

Face recognition is one of the well studied problems by researchers in computer visions. Among the challenges of this task are the occurrence of different facial expressions like happy or sad, and different views of the images such as front and side views. This paper experiments a publicly available dataset that consists of 200,000 images of celebrity faces. Deep Learning technique is gaining its popularity in computer vision and this paper applies this technique for face recognition problem. One of the techniques under deep learning is Convolutional Neural Network (CNN). There is also pre-trained CNN models that are AlexNet and GoogLeNet, which produce excellent accuracy results. The experimental results indicate that AlexNet is better than basic CNN and GoogLeNet for face recognition.

## **Introduction**

Image recognition and comparison is a topic within computer science that still is partially problematic. No algorithm is perfect and there is still no way to solve everything with one algorithm. Lifesum is a company that is interested in this field, they want a solution that can be used within their distribution of calorie counting applications. Our task is to research a solution that could work when classifying images of food. This will be done by finding a previously created solution and adapt it. This is because it is too complex to create our own algorithm within ten weeks. It is not necessary that the classification is perfect, but it should work well enough so that Lifesum find it useful in production.

Lifesum has thought about a solution to the problem within their calorie counting applications. They want their users to be able to take a picture of a meal and store data associated to that picture. Then later take a new picture of the same kind of meal and retrieve the stored data. Solving this problem comes down to the following research questions:

## **Purpose**

Lifesum has come to the conclusion that the manual way to enter a meal's ingredients today is not ideal. They would like a new method where users take a picture of their food and the application answers with a possible set of meals. The objective is not to create a well functional solution, but to researching how Lifesum could proceed to achieve this as a future goal. It is also important for them to see how far the image processing field has come. Researching for a future solution, Lifesum hopes to make the user experience better. With a better user experience they expect to extend their user base and generate an increased revenue.

## **Machine Learning**

Artificial intelligence is one field that is using machine learning. Machine learning is a science discipline that focuses on creating algorithms that can learn from data [12]. This is interesting since we will need to use algorithms that make use of this to make image recognition usable [4]. This could also be used to boost performance of algorithms.

## **Recognition**

Within computer vision there are several fields. The most interesting one out of these for this thesis is recognition. Recognition is when you make a computer study an image in such a way that it can determine what is in the picture. If this could be used within this thesis we could determine what food the user actually is eating which would make our task easier. When creating such a system it is necessary to implement a machine learning algorithm [4]. Recognition is much more than just determining the objects in images. It is also used when identifying preregistered fingerprints [23] and faces in authorization systems and used when trying to detect predetermined conditions in images.

## **Libraries used**

### **Tensor flow**

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications.

### **Keras**

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation

### **Matplotlib**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

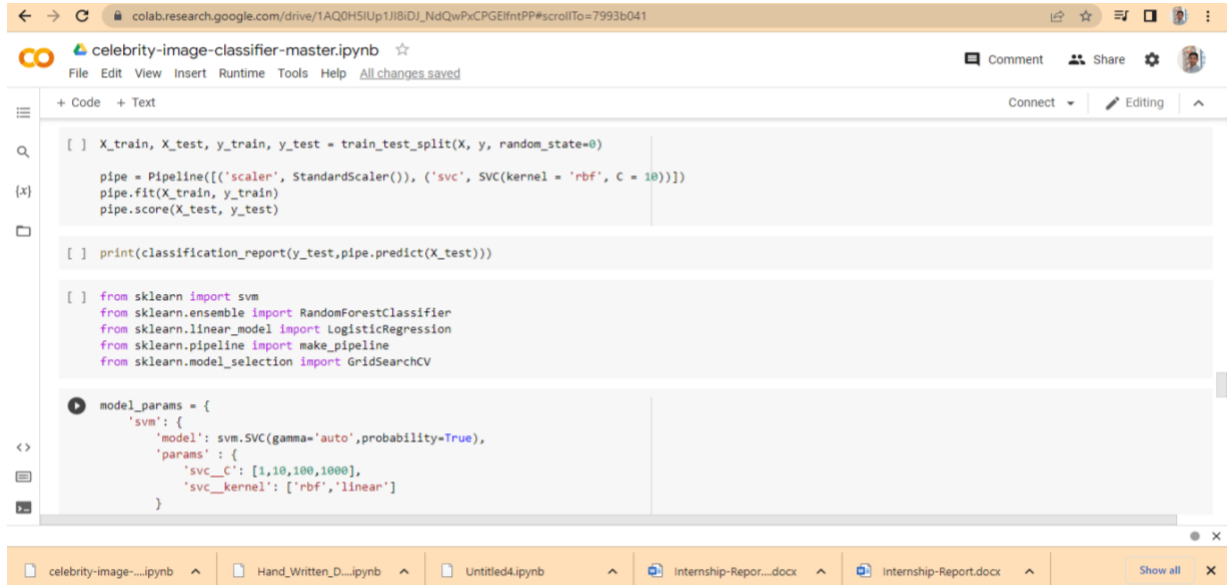
### **Pyplot**

[matplotlib.pyplot\(\)](#) is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

### **PIL**

PIL is a library of advanced image tools having full name Pillow. It is used for noise cancellation and to draw modified pixels by noise. It is useful to Image crop and various subjectal techniques.

# Source code



The screenshot shows a Google Colab notebook interface. The browser address bar displays the URL: `colab.research.google.com/drive/1AQ0H5IUpl1Jl8IDJ_NdQwPxCPGEfntPP#scrollTo=7993b041`. The notebook title is "celebrity-image-classifier-master.ipynb". The code editor contains the following Python code:

```
[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

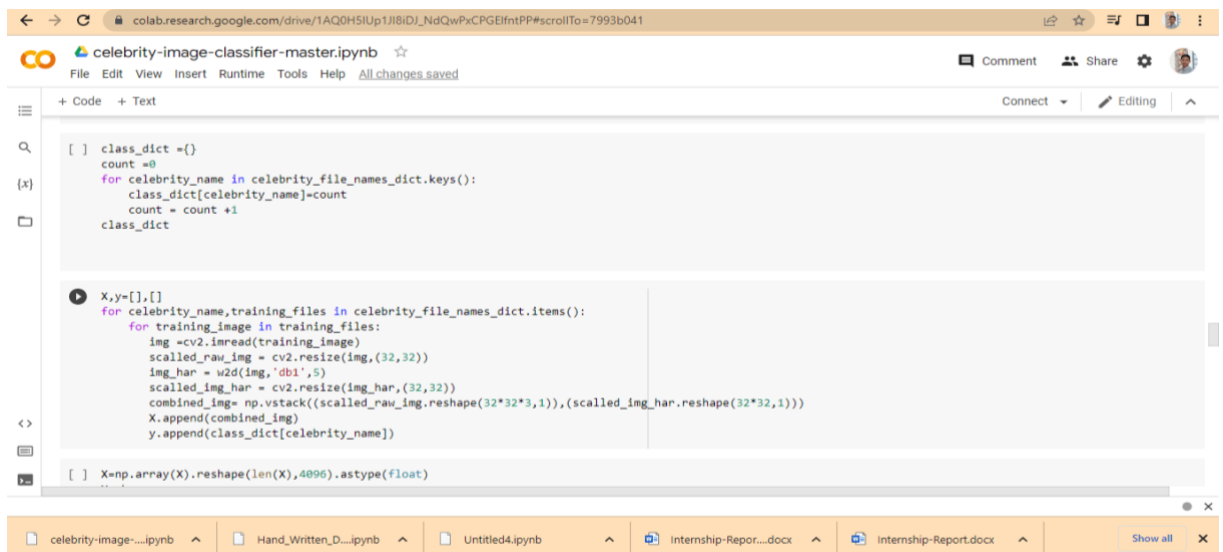
pipe = Pipeline([('scaler', StandardScaler()), ('svc', SVC(kernel = 'rbf', C = 10))])
pipe.fit(X_train, y_train)
pipe.score(X_test, y_test)

[ ] print(classification_report(y_test, pipe.predict(X_test)))

[ ] from sklearn import svm
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import GridSearchCV

model_params = {
    'svm': {
        'model': svm.SVC(gamma='auto', probability=True),
        'params' : {
            'svc__C': [1,10,100,1000],
            'svc__kernel': ['rbf', 'linear']
        }
    }
```

The bottom of the notebook shows a tab bar with several open files: "celebrity-image-...ipynb", "Hand\_Written\_D...ipynb", "Untitled4.ipynb", "Internship-Report....docx", and "Internship-Report.docx".



The screenshot shows the same Google Colab notebook interface. The code editor contains the following Python code:

```
[ ] class_dict = {}
count = 0
for celebrity_name in celebrity_file_names_dict.keys():
    class_dict[celebrity_name] = count
    count = count + 1
class_dict

X,y=[],[]
for celebrity_name, training_files in celebrity_file_names_dict.items():
    for training_image in training_files:
        img = cv2.imread(training_image)
        scaled_raw_img = cv2.resize(img, (32, 32))
        img_har = w2d(img, 'db1', 5)
        scaled_img_har = cv2.resize(img_har, (32, 32))
        combined_img = np.vstack((scaled_raw_img.reshape(32*32*3, 1), (scaled_img_har.reshape(32*32, 1))))
        X.append(combined_img)
        y.append(class_dict[celebrity_name])

[ ] X = np.array(X).reshape(len(X), 4096).astype(float)
```

The bottom of the notebook shows the same tab bar as the previous screenshot, with files: "celebrity-image-...ipynb", "Hand\_Written\_D...ipynb", "Untitled4.ipynb", "Internship-Report....docx", and "Internship-Report.docx".

celebrity-image-classifier-master.ipynb

```
celebrity_file_name_dict = {}
for img_dir in img_dirs:
    celebrity_name = img_dir.split('/')[-1]
    celebrity_file_name_dict[celebrity_name] = []
    for entry in os.scandir(img_dir):
        roi_color = get_cropped_image_if_2_eyes(entry.path)
        if roi_color is not None:
            cropped_folder = path_to_cr_data + celebrity_name
            if not os.path.exists(cropped_folder):
                os.makedirs(cropped_folder)
            cropped_img_dirs.append(cropped_folder)
            print("Generating cropped images in folder: ", cropped_folder)
            cropped_file_name = celebrity_name + str(count) + ".png"
            cropped_file_path = cropped_folder + "/" + cropped_file_name
            cv2.imwrite(cropped_file_path, roi_color)
            celebrity_file_name_dict[celebrity_name].append(cropped_file_path)
            count += 1

[ ] celebrity_file_names_dict = {}
for img_dir in cropped_image_dirs:
    celebrity_name = img_dir.split('/')[-1]
    file_list = []
```

colab.research.google.com/drive/1AQ0H5iUp1Jl8Dj\_NdQwPxCPGEfntPP#scrollTo=7993b041

celebrity-image-classifier-master.ipynb

```
[ ] original_img = cv2.imread('F:/sharath files/project internship/Sports-image-classifier-master/model/test_images/sharpova1.jpg')
plt.imshow(original_img)

[ ] cropped_img = cv2.imread('F:/sharath files/project internship/Sports-image-classifier-master/model/test_images/sharpova2.jpg')
plt.imshow(cropped_img)

cropped_img_no_2_eyes = get_cropped_image_if_2_eyes('F:/sharath files/project internship/Sports-image-classifier-master/model/test_images/sharpova2.jpg')
cropped_img_no_2_eyes

[ ] path_to_data = 'F:/sharath files/project internship/Sports-image-classifier-master/model/dataset/'
path_to_cr_data = 'F:/sharath files/project internship/Sports-image-classifier-master/model/dataset/cropped/'

[ ] import os
img_dirs = []
for entry in os.scandir(path_to_data):
    if entry.is_dir():
        img_dirs.append(entry.path)

[ ] img_dirs
```

colab.research.google.com/drive/1AQ0H5Up1Jl8lDJ\_NdQwPxCPGEfntPP#scrollTo=7993b041

celebrity-image-classifier-master.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
[ ] # reconstruction
    imArray_H=pywt.waverec2(coeffs_H, mode);
    imArray_H *= 255;
    imArray_H = np.uint8(imArray_H)

[ ] return imArray_H

im_har = w2d(cropped_img,'db1',5)
plt.imshow(im_har,cmap='gray')

def get_cropped_image_if_2_eyes(image_path):
    img = cv2.imread(image_path)
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray,1.3, 5)
    for (x,y,w,h) in faces:
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(roi_gray)
        if len(eyes) >= 2:
            return roi_color
```

celebrity-image-...ipynb Hand\_Written\_D...ipynb Untitled4.ipynb Internship-Repor...docx Internship-Report.docx Show all

colab.research.google.com/drive/1AQ0H5Up1Jl8lDJ\_NdQwPxCPGEfntPP#scrollTo=7993b041

celebrity-image-classifier-master.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

```
[ ] im_har = w2d(cropped_img,'db1',5)
    plt.imshow(im_har,cmap='gray')

def get_cropped_image_if_2_eyes(image_path):
    img = cv2.imread(image_path)
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        roi_gray = gray

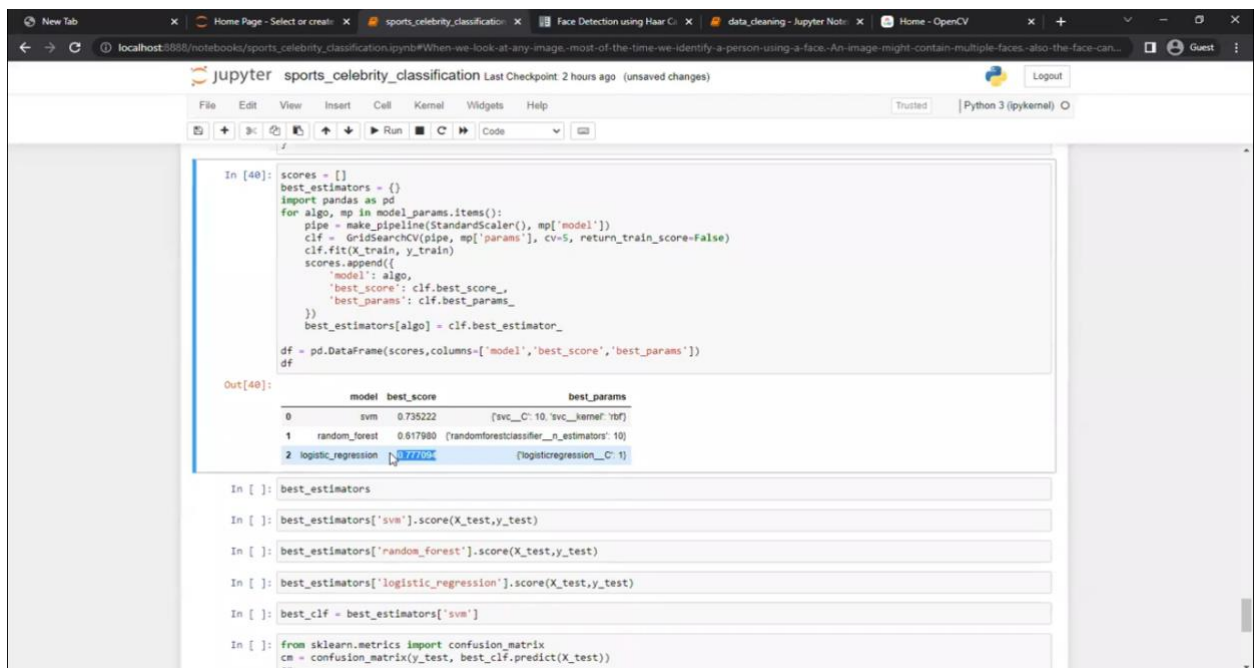
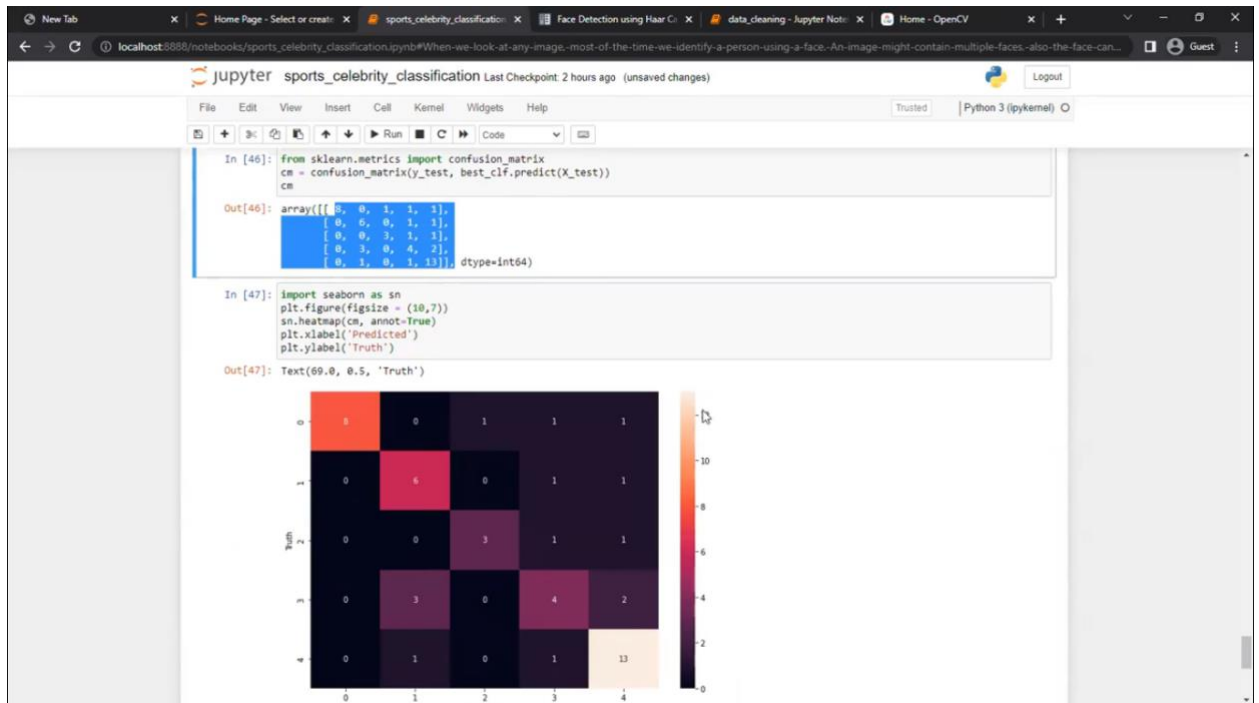
import numpy as np
import pywt
import cv2

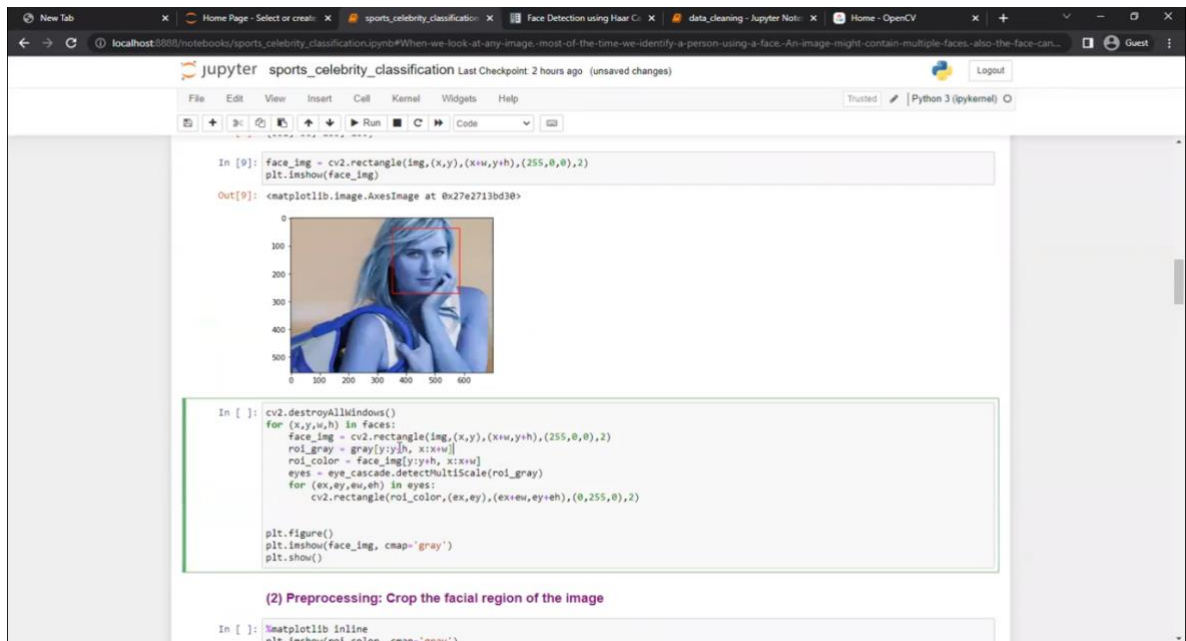
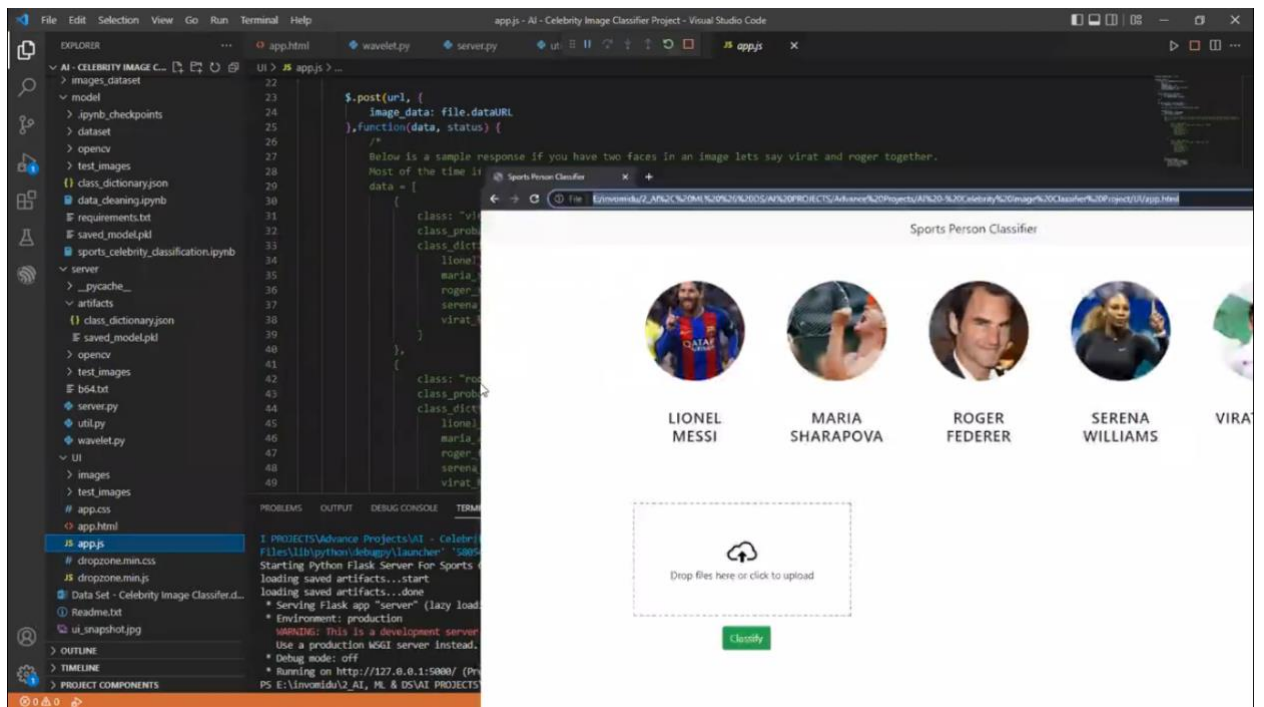
def w2d(img, mode='haar', level=1):
    imArray = img
    #Datatype conversions
    #convert to grayscale
    imArray = cv2.cvtColor( imArray,cv2.COLOR_RGB2GRAY )
    #convert to float
    imArray = np.float32(imArray)
    imArray /= 255;
```

celebrity-image-...ipynb Hand\_Written\_D...ipynb Untitled4.ipynb Internship-Repor...docx Internship-Report.docx Show all



# Output





## **Technologies used in this project**

- Python
- Numpy and OpenCV for data cleaning
- Matplotlib & Seaborn for data visualization
- Sklearn for model building Jupyter notebook, visual studio code and pycharm as IDE
- Python flask for http server
- HTML/CSS/Javascript for UI

## **Learning after internship**

While doing the internship I got a chance to explore many machine learning algorithms like KNN, Linear regression etc. In his project I had a chance to explore neural networks more specifically Convolutional Neural Network. I learnt about the working of neural networks and also could explore tinker library which helped me in constructing a canvas for giving handwritten digit as the input.

## **Conclusions**

The analysis has given results that will be explained and discussed in this chapter. The problem definition will be reflected upon and there will be explanations about what has been done to answer the questions asked in the problem definition. When these questions are answered, information will be given about what could be improved and what restrictions our project has been limited by.

When studying the field, the solutions found were either trivial and not accurate enough, or accurate but too complex, requiring more time for implementation and training a keypoint detection algorithm would have been necessary for better results. If Lifesum would use a keypoint detection algorithm and use a training algorithm to improve the detection, it could potentially be working good enough to be used in production. It should be noted that Adaboost and multi-classification possibly also needs to be implemented for a successful implementation.

Finding the best matching image within a set of images is the best option. This is because it is too general and not accurate enough to recognize objects within images. When implementing an image recognition algorithm, you will need to train it for every object you want to recognize. This would mean that if Lifesum chooses this solution, they will need to gather several images for every meal they want to be able to recognize. These images also need to be for commercial use. This leads us to the assumption that image recognition probably is not the most optimal solution for Lifesum.

## **Limitations**

The complexity of the image recognition field has prevented us from creating the best possible solution. Our previous low knowledge within image processing was definitely a limitation when creating this thesis. Information found within this field is heavily based on theoretical math and machine learning, which made this project rather difficult. Both of these fields can be studied as master programs and can get much harder when digging deeper into them.

## **Hardware Requirements**

**Processor:** intel CORE i5

**Hard Disk :** 512GB

**RAM:** 4GB

## **Software Requirements**

**Operating System:** Windows 7, Linux, MAC

**Language:** Python 3

## **Bibliography**

[1] Adobe Systems. Adobe Photoshop. url: [http : // www . photoshop . com/](http://www.photoshop.com/).

[2] Adrian Rosebrock. How-To: 3 Ways to Compare Histograms using OpenCV and Python – PyImage Search. url: [http://www.pyimagesearch. com/2014/07/14/3- ways-compare- histograms- using- open cvpython /](http://www.pyimagesearch.com/2014/07/14/3-ways-compare-histograms-using-open-cvpython/) (visited on 04/15/2015).

[3] Alexander Mordvintsev and Abid K. Introduction to SIFT (ScaleInvariant Feature Transform). url: [https://opencv-python-tutroals. readthedocs.org/en/latest/py\\_tutorials/py\\_feature2d/py\\_ sift\\_intro/py\\_sift\\_intro.html](https://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html) (visited on 06/11/2015).

[4] Henrik Andreasson. “Object Recognition using Vision and Machine Learning”. url: [http://aass.oru.se/~tdt/ml/extra-slides/ML\\_obj\\_rec.pdf](http://aass.oru.se/~tdt/ml/extra-slides/ML_obj_rec.pdf).

[5] Mike Bostock. D3.js - Data-Driven Documents. url: [http://d3js. org/](http://d3js.org/) (visited on 05/13/2015).

[6] Gary Bradski and Adrian Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library. "O'Reilly Media, Inc.", Sept. 24, 2008, pp. 201–206. 579 pp. isbn: 9780596554040.

[7] CBS Interactive Inc - Danny Gallagher. Google cooking up AI to count calories in your food photos - CNET. url: <http://www.cnet.com/news/google-working-on-ai-that-can-count-calories-in-your-food-photos/> (visited on 06/09/2015).



