Face Recognition

Made by Simarjit Singh Bains

GitHub Repository for “Choose File”: <https://github.com/Simar2906/Face-Recognizer>

Selfmade walkthrough, How to run code: <https://www.youtube.com/watch?v=9Qe2hFba5XE&t=63s>

GitHub Repository for “Youtube URL”: pending

# Problem Statement

# 

# Index

|  |  |
| --- | --- |
| S no. | Title |
| 1 | [What I learnt from this Assignment](#_What_I_learnt) |
| 2 | [What I could have Done Better](#_What_I_could) |
| 3 | [Steps taken by me for the Project](#_Steps_taken_by) |
| 4 | [Qualities of the Delivered System](#_Qualities_of_delivered) |
| 5 | [Flask Practice](#_Flask_Practice) |
| 6 | [Training the Deep Learning Model](#_Training_the_Deep) |
| 7 | [Development of Application](#_Development_of_Application) |
| 8 | [Trying to Host on Heroku](#_Trying_to_Host) |
| 9 | [Using python-vlc](#_Using_python-vlc) |

# What I learnt from this Assignment

* Revised basics of facial recognition, image augmentation, transfer learning, deep learning and web development
* Learned about Restful APIs and their implementation
* Gained experience in ML and DL model deployment with Github Pages and Heroku
* Learned how to server GET and POST requests
* Learned about many libraries like python-vlc, imageio etc

# What I could have done better

* Was not able to deploy model on Heroku. Having more knowledge about Heroku and model deployment could have averted this problem.
* Didnt have time left to develop deliverable with “youtube url and time stamp” input. But have provided proof of concept on how it could be done.
* Could have written cleaner code, with functions performing different tasks.

# Steps taken by me for the Project

* The problem requires a system that upon input recognizes whether a person is present between the time stamps for the provided YouTube video.
* Revised Concepts of facial recognition from [Here](https://towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2)
* I first thought of practicing with Flask API on a simple past machine learning project I made
* After this I trained my deep learning model on a dataset of celebrity faces which gave and accuracy of **77.75%**
* I started development of a python file which will use Flask API to provide a frontend to my model which would request image from a choose file method.
* Upon completion of the project I thought of deploying it on Heroku. The built project was giving an error so gave up on deployment.
* Started working with python-vlc to integrate YouTube input for my application. (currently incomplete)

# Qualities of delivered system

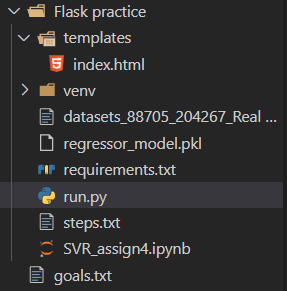
* Deep learning model trained on **14k** images with accuracy of **77.75%**
* Use of Flask API to create a frontend for deployment
* Model has ability to classify the provide image of the celebrity
* Model has ability to detect a non-Celebrity Face
* Model has ability to predict if no face is present in the provided image

# 

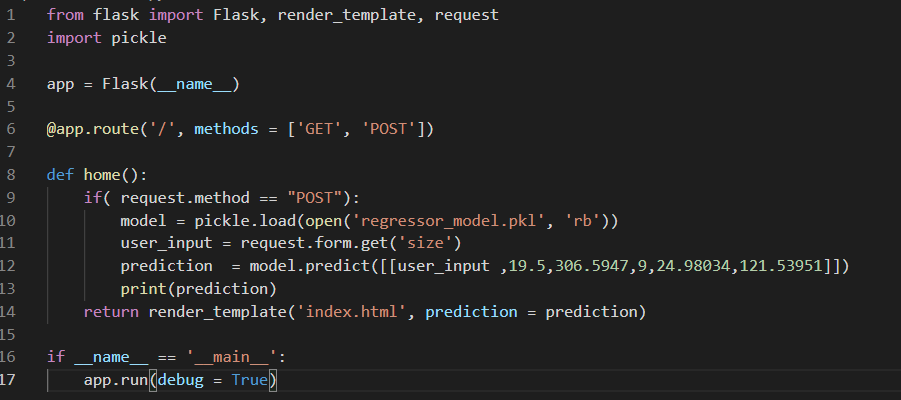
# Flask Practice

* Used an older machine learning project to get grasp over flask.
* Used pickle to save my model, and created an application file

Folder Structure



Run.py



## References:

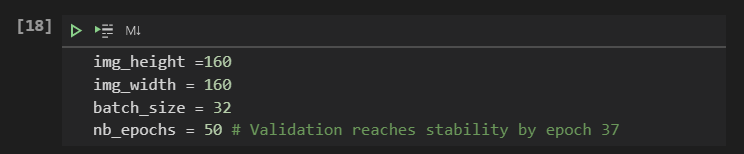
Building API for Machine Learning Model with Flask: <https://www.youtube.com/watch?v=tle5A_vpdTE>

Serving a Machine Learning Model Via REST API

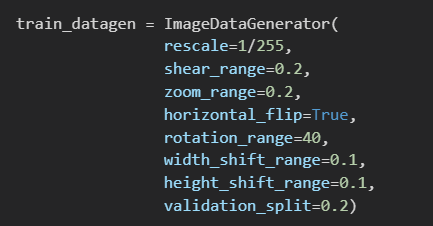
<https://towardsdatascience.com/serving-a-machine-learning-model-via-rest-api-5a4b38c02e90>

# Training the Deep Learning Model

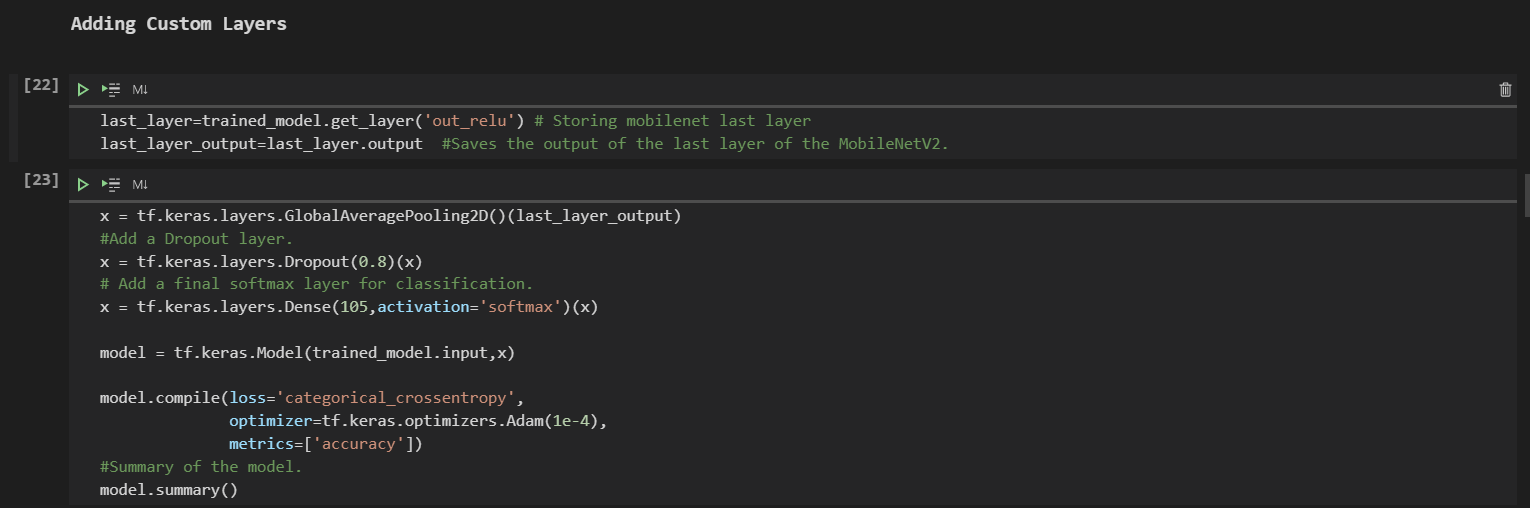
* Dataset Used: 105 Celebrity from pinterest on Kaggle  
  <https://www.kaggle.com/hereisburak/pins-face-recognition>
* Used Keras, OpenCV and transfer learning with MobileNetV2 with ‘imagenet’ competition weights
* Image Dimensions

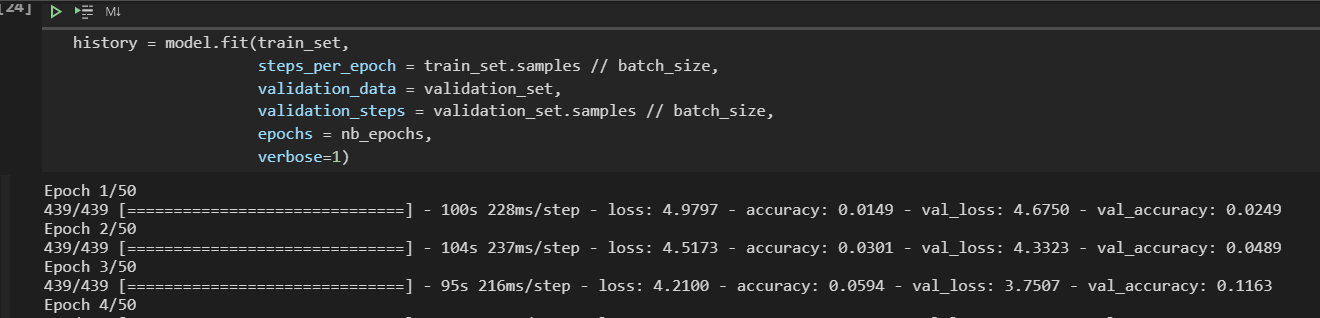


* Image augmentation

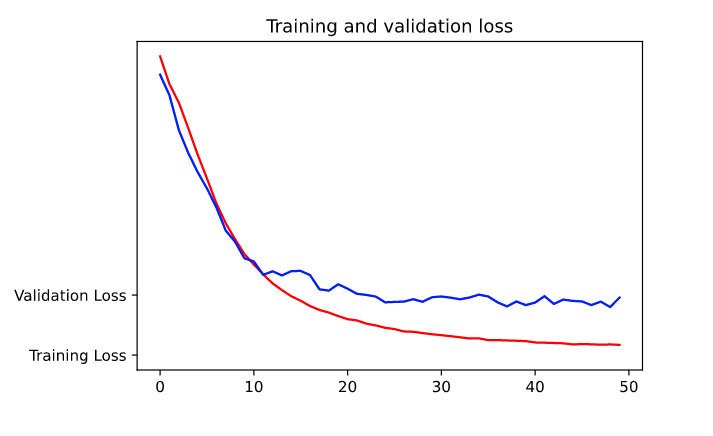


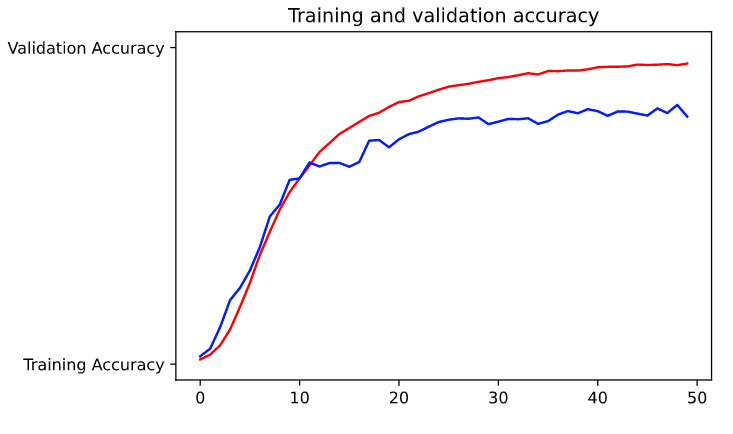
* Trained the model for 1.5 hours for 50 epochs on the dataset





* Model was able to give 77.75% validation accuracy upon completion. Saved the models in a separate directory



****

## References

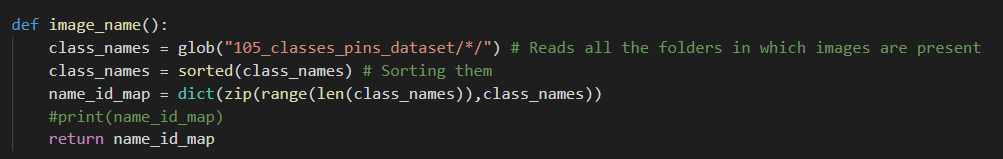
Kaggle Notebook preforming Recognition analysis: <https://www.kaggle.com/srikeshram/celebrity-face-recognition>

Rest Code used from my own repository:

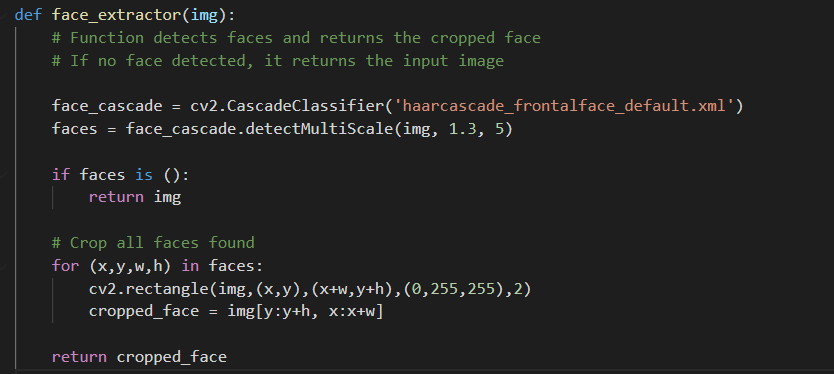
<https://github.com/Simar2906/English-Handwritten-Characters-Recognition>

# Development of Application

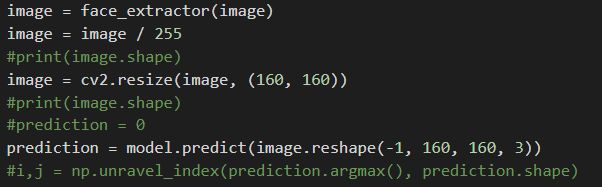
* After studying about flask and deep learning, I started development of the application file.
* Created a function to create a dictionary from the dataset directory to return the class name to html file.



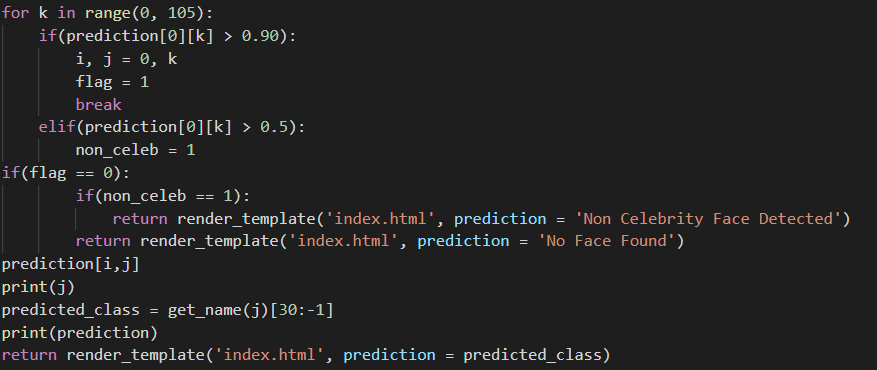
* Wrote code for a “POST” request, which will save the inputted image
* Used a face\_extractor function to crop out the face with help of ‘haarcascade\_frontalface\_default.xml’



* Applied image resizing and called loaded model to predict the class of the inputted image



* Created a loop to determine whether the provided image is of a celebrity, a non-celebrity, or if a face is not found, used a get\_name function to return the class name from the previously created dictionary



## References

Deployment of Deep Learning Model using Flask: <https://www.youtube.com/watch?v=CSEmUmkfb8Q>

Quick Image Classifier Web Application with Flask, Keras and Bokeh:

<https://stackschool.io/quick-image-classifier-web-application-with-flask-keras-and-bokeh/>

Deploying a Machine Learning Model as a REST API

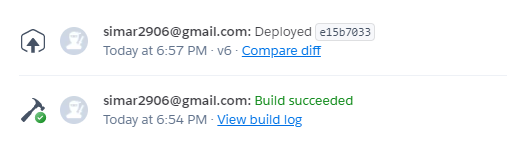
<https://towardsdatascience.com/deploying-a-machine-learning-model-as-a-rest-api-4a03b865c166>

Deploying Keras Deep Learning Models with Flask

<https://towardsdatascience.com/deploying-keras-deep-learning-models-with-flask-5da4181436a2>

# Trying to Host on Heroku

* Learning basics of Heroku and created relevant files
* Successfully uploaded and built the application



<https://simar2906-face-recognition.herokuapp.com/>

* Discovered some issue that caused the application to keep on crashing, so put deployment on hold

# Using python-vlc

* I came across this library for python which has support for VLC Media Player functionalities.
* VLC has the capability to play youtube videos and skip to time stamps which would have made it easy for me to navigate through the video.
* This library also has the screenshot functionality, which would have helped me to get screenshots at some time intervals between the given time stamps

Steps:

1. Use html form to get url and time stamps from the user
2. Use vlc (library) to play the video and take and store screenshots in between the time stamps provided at every 3 seconds
3. Apply the already built predict function on the images to detect whether there is a celebrity in the video or not, or if there is a non-celebrity or even if there is no face present in the video

## References

[Using python-vlc to get screenshots from youtube video](https://stackoverflow.com/questions/52918532/using-python-vlc-to-get-screenshots-from-youtube-video)

<https://stackoverflow.com/questions/52918532/using-python-vlc-to-get-screenshots-from-youtube-video>