Face Recognition

Made by Simarjit Singh Bains

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

GitHub Repository for “Youtube URL”:

# Problem Statement

# 

# Index

|  |  |  |
| --- | --- | --- |
| S no. | Title | Page |
| 1 | [Steps taken by me for the Project](#_Steps_taken_by) | 2 |
| 2 | [Flask Practice](#_Flask_Practice) | 3 |
| 3 | [Training the Deep Learning Model](#_Training_the_Deep) | 4 |
| 4 | [Development of Application](#_Development_of_Application) | 6 |
| 5 | [Trying to Host on Heroku](#_Trying_to_Host) | 8 |
| 6 | [Using python-vlc](#_Using_python-vlc) | 9 |

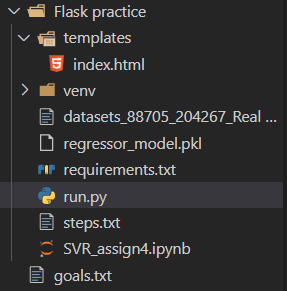
# Steps taken by me for the Project

* 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)

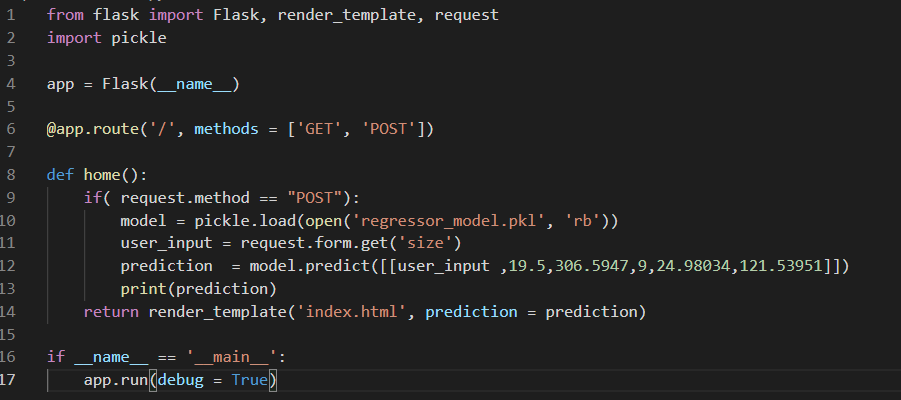
# Flask Practice

* 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:

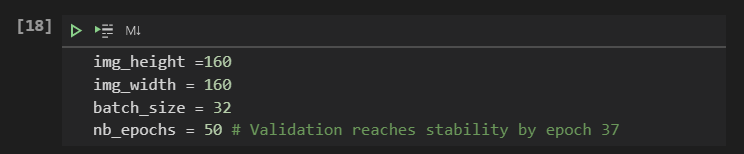
Building API for Machine Learning Model with Flask: <https://www.youtube.com/watch?v=tle5A_vpdTE>

Serving a Machine Learning Model Via REST API

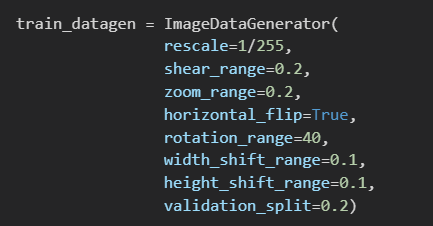
<https://towardsdatascience.com/serving-a-machine-learning-model-via-rest-api-5a4b38c02e90>

# Training the Deep Learning Model

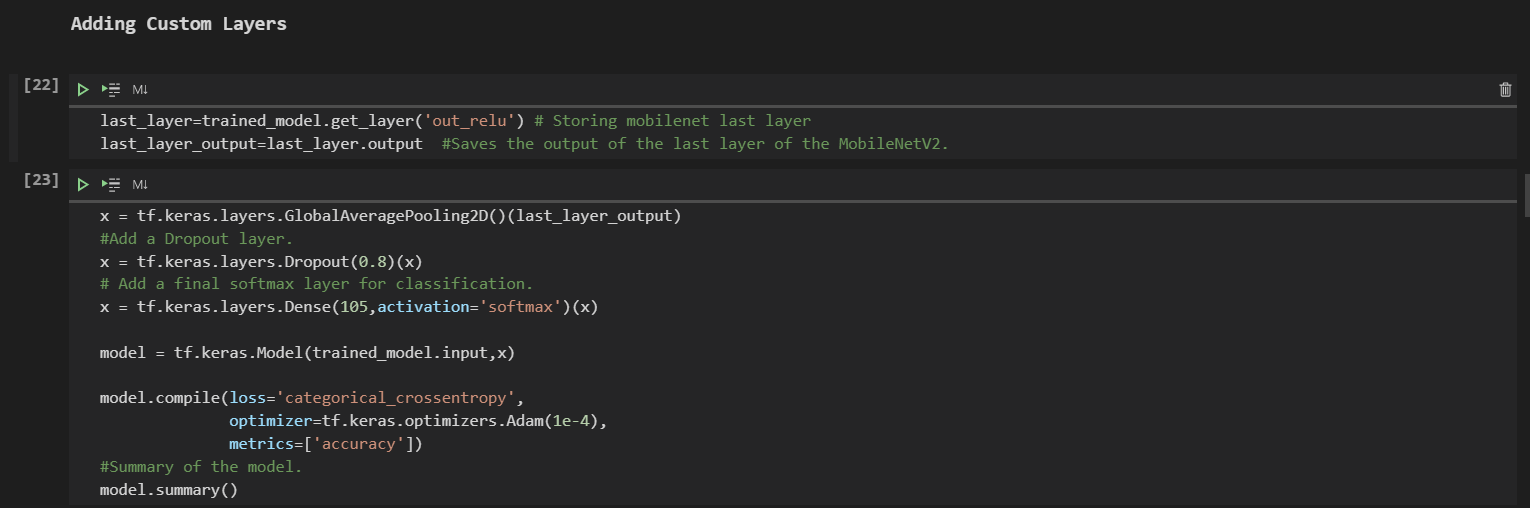
* Dataset Used: 105 Celebrity from pinterest on Kaggle  
  <https://www.kaggle.com/hereisburak/pins-face-recognition>
* Used Keras, OpenCV and transfer learning with MobileNetV2 with ‘imagenet’ competition weights
* Image Dimensions

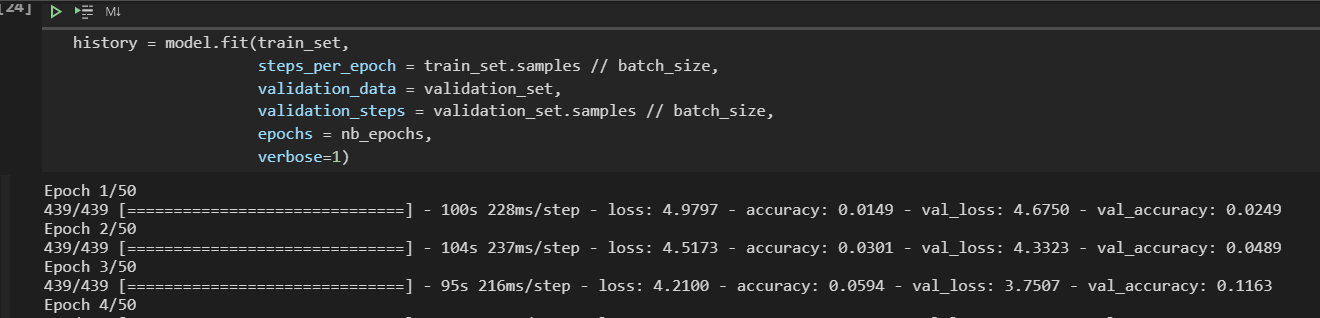


* Image augmentation

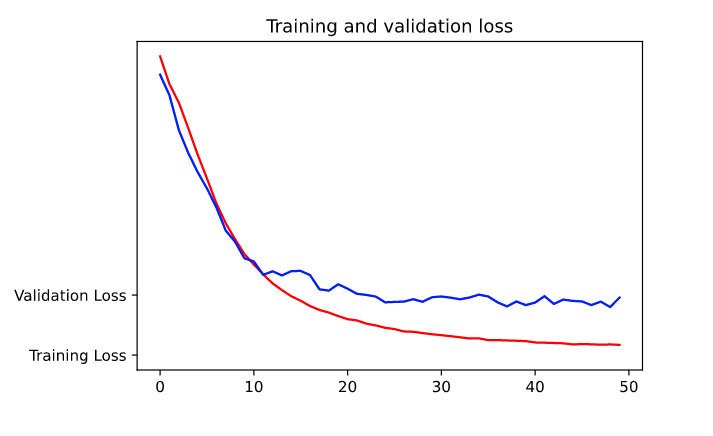


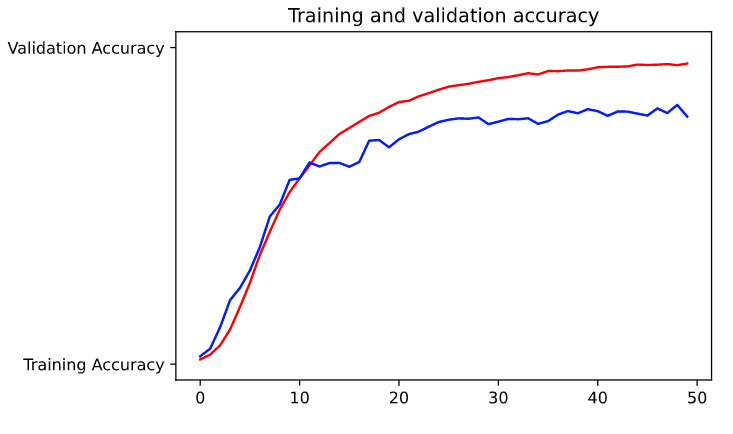
* Trained the model for 1.5 hours for 50 epochs on the dataset





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



****

## References

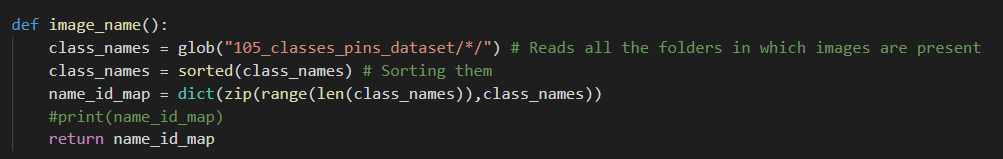
Kaggle Notebook preforming Recognition analysis: <https://www.kaggle.com/srikeshram/celebrity-face-recognition>

Rest Code used from my own repository:

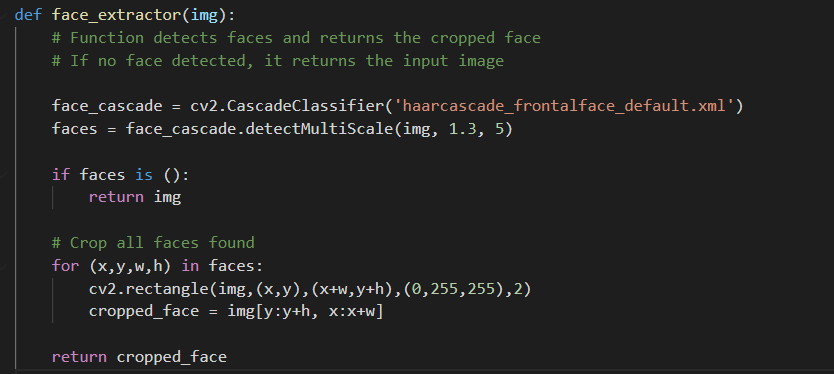
<https://github.com/Simar2906/English-Handwritten-Characters-Recognition>

# Development of Application

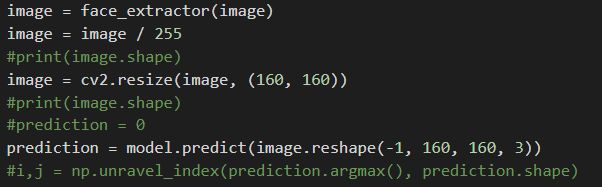
* After studying about flask and deep learning, I started development of the application file.
* Created a function to create a dictionary from the dataset directory to return the class name to html file.



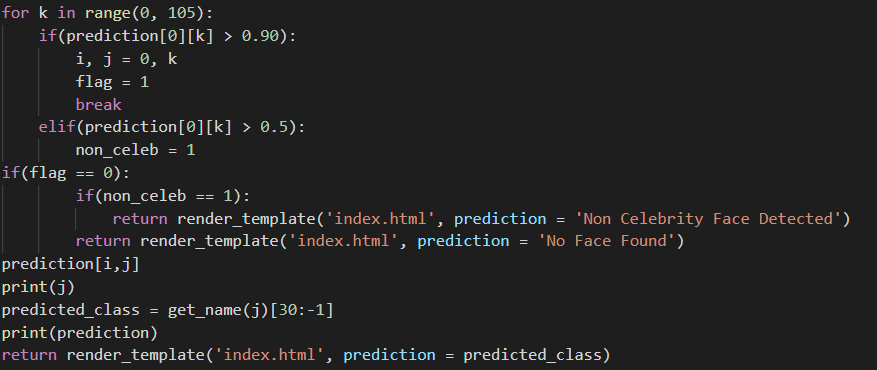
* Wrote code for a “POST” request, which will save the inputted image
* Used a face\_extractor function to crop out the face with help of ‘haarcascade\_frontalface\_default.xml’



* 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

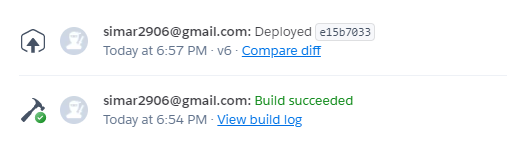
<https://towardsdatascience.com/deploying-a-machine-learning-model-as-a-rest-api-4a03b865c166>

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