Sports
Person Image Classifier

Introduction

- Combining ML + CV + Web Dev
- Using openCV managed various Computer vision for various tasks like:
    - **Image preprocessing which involved resizing of the image**
    - **Normalizing**
    - **Augmenting the data to enhance the training set diversity and robustness**
- In the machine learning phase, I experimented with SVM, Random Forest, and Logistic Regression models, tuning their hyperparameters for optimal performance.
- Evaluating the models involved assessing their accuracy and performance metrics to ensure the best outcomes.
- Finally completed the work with **flask server** which is still in testing purpose.

Problem statement

- Develop an automated sports celebrity recognition system using advanced computer vision and machine learning for real-time image classification and a user-friendly web application.

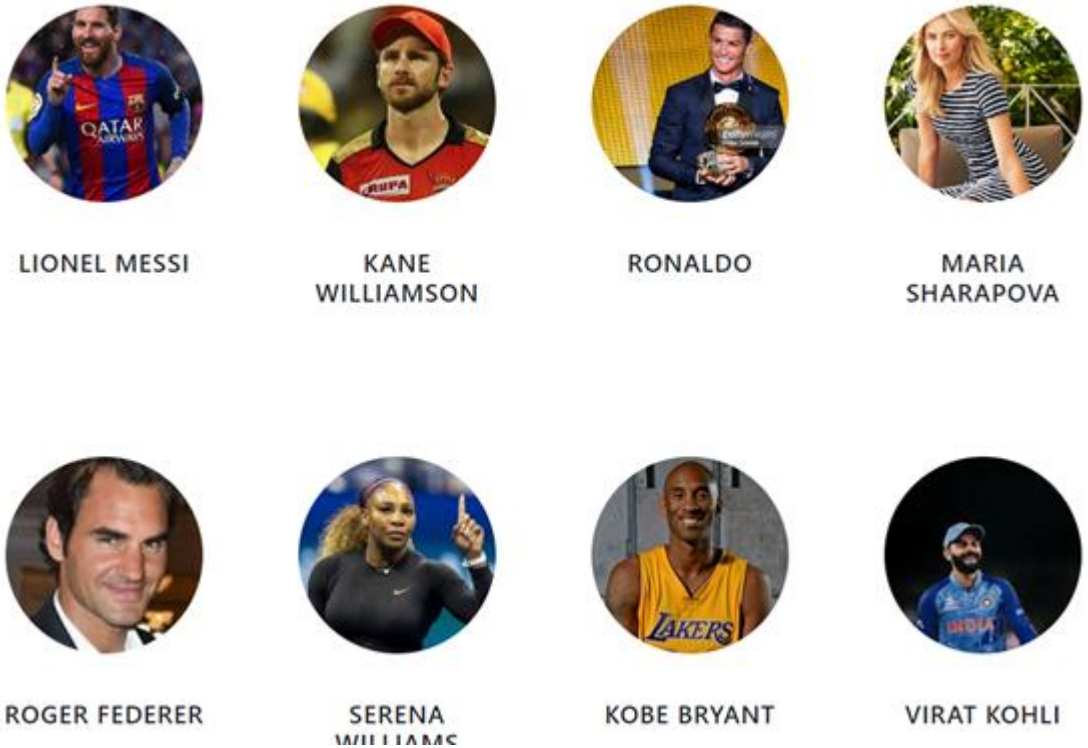
Objectives

1) **Image Preprocessing:** Implement robust image preprocessing techniques, including resizing, normalizing, and augmenting, to enhance the quality and diversity of the training dataset.
2) **Model Training and Optimization:** Experiment with and fine-tune multiple machine learning models, such as SVM, Random Forest, and Logistic Regression, to determine the best-performing model for sports celebrity image classification.
3) **Performance Evaluation:** Conduct thorough evaluation of the trained models using appropriate performance metrics to ensure high accuracy and reliability in predictions.
4) **Real-Time Prediction:** Develop a user-friendly web application using Flask that allows users to upload images and receive real-time predictions of the sports celebrity depicted in the image.
5) **Seamless Integration:** Ensure a seamless integration of computer vision, machine learning, and web development to provide an interactive and efficient user experience.

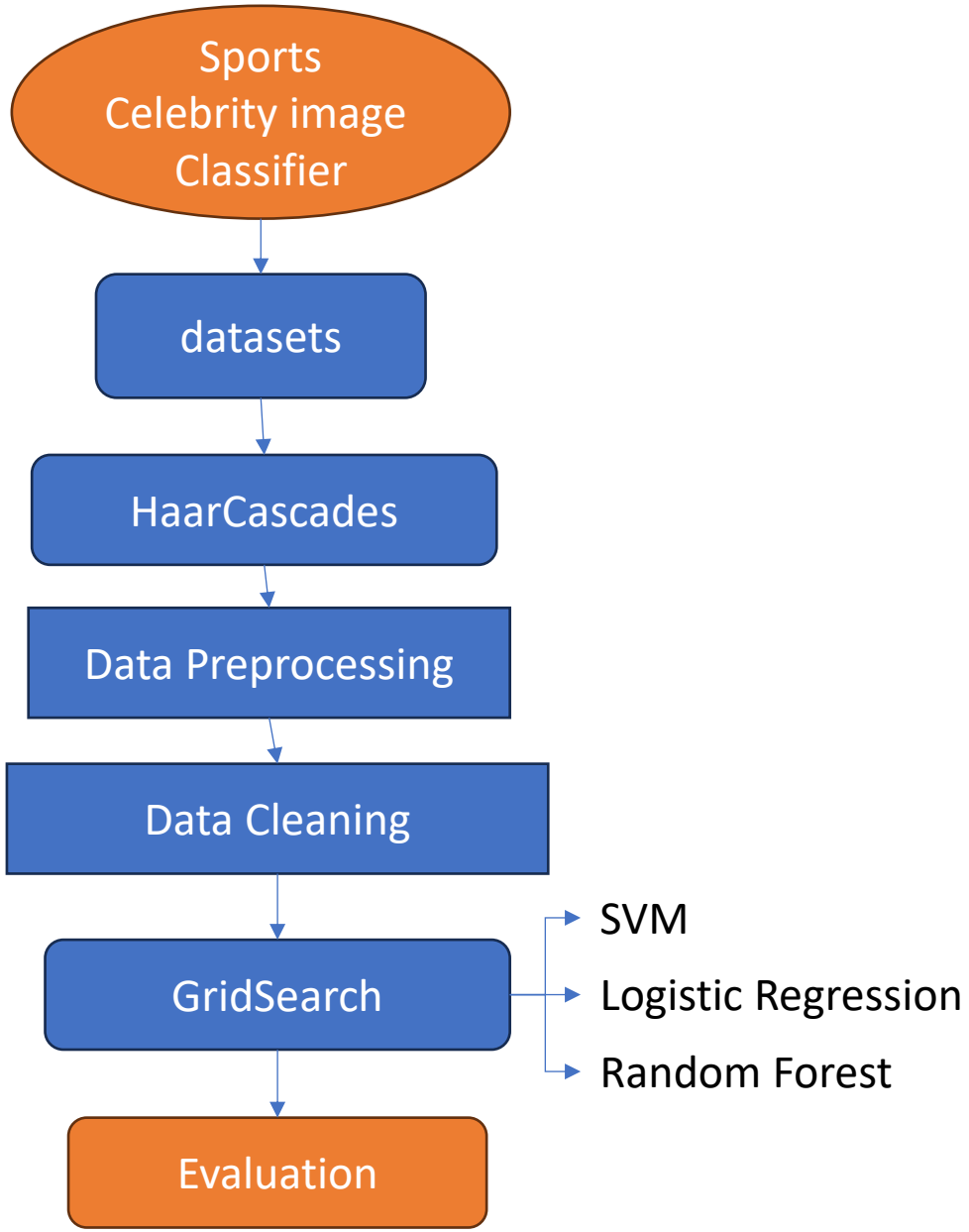
Dataset Description

Dataset is taken from:

- Kaggle - https://www.kaggle.com/datasets/yaswanthgali/sport-celebrity-image-classification
- Other resources
  - Hugging Face
  - Git



Flowchart


Sports
Celebrity image
Classifier
datasets
HaarCascades
Data Preprocessing
Data Cleaning
GridSearch
SVM
Logistic Regression
Random Forest
Evaluation

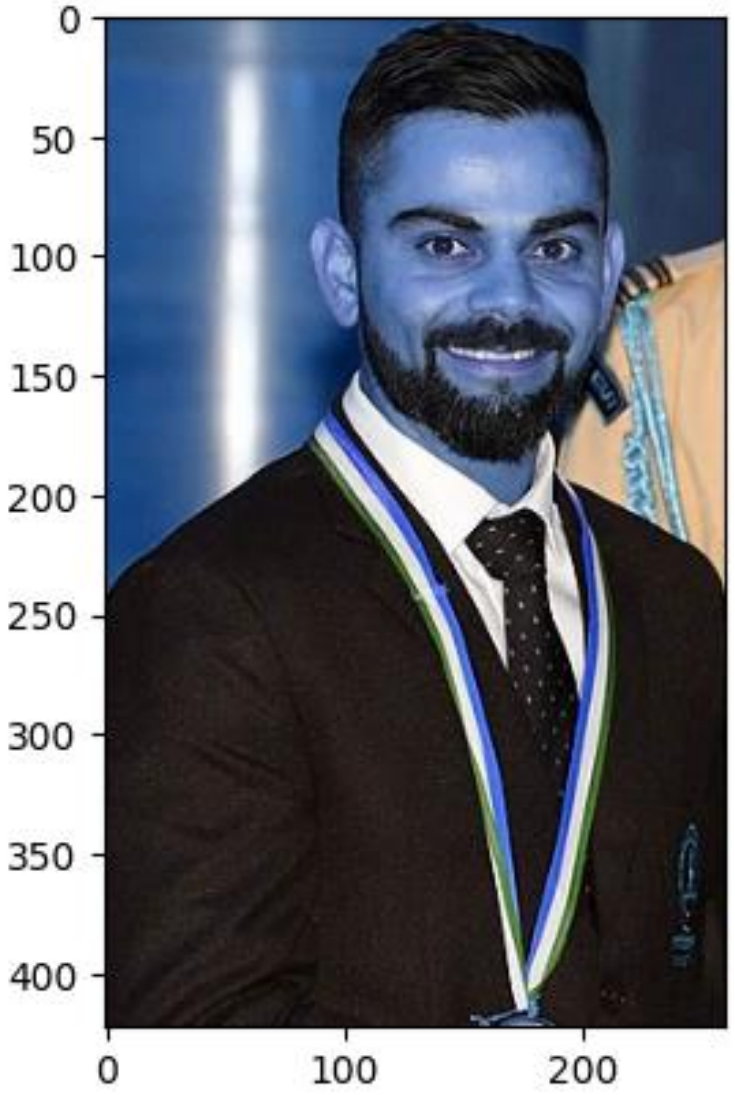
HaarCascades

- HaarCascades are used for detecting faces, eyes, and various objects in images.
- This method is highly efficient in identifying specific features within an image, making it ideal for tasks requiring precise object recognition.
- By employing Haar cascades, we can accurately detect and focus on regions of interest, which significantly improves the quality and relevance of the data used for training our machine learning models.
- Additionally, Haarcascades are widely utilized due to their robustness and speed, making them suitable for real-time applications.

Why not CNN only Haarcascades

I chose to use Haar cascades over CNNs for the initial face and eye detection stage primarily due to the following reasons:

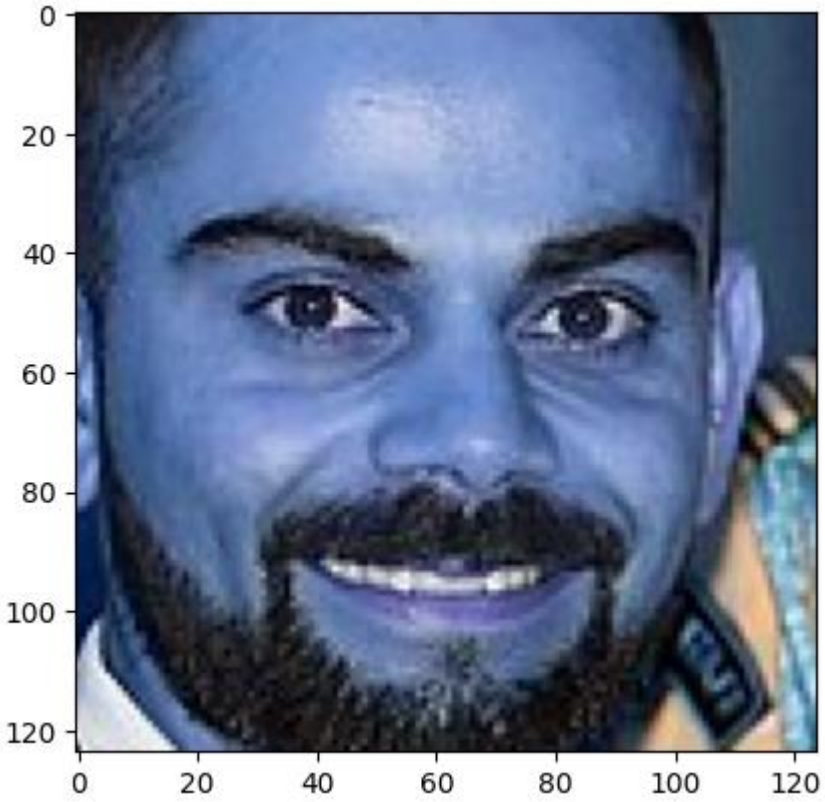
- **Speed and Effciency**
- **Simplicity**
- **Resource Management**
- **Historical performance**



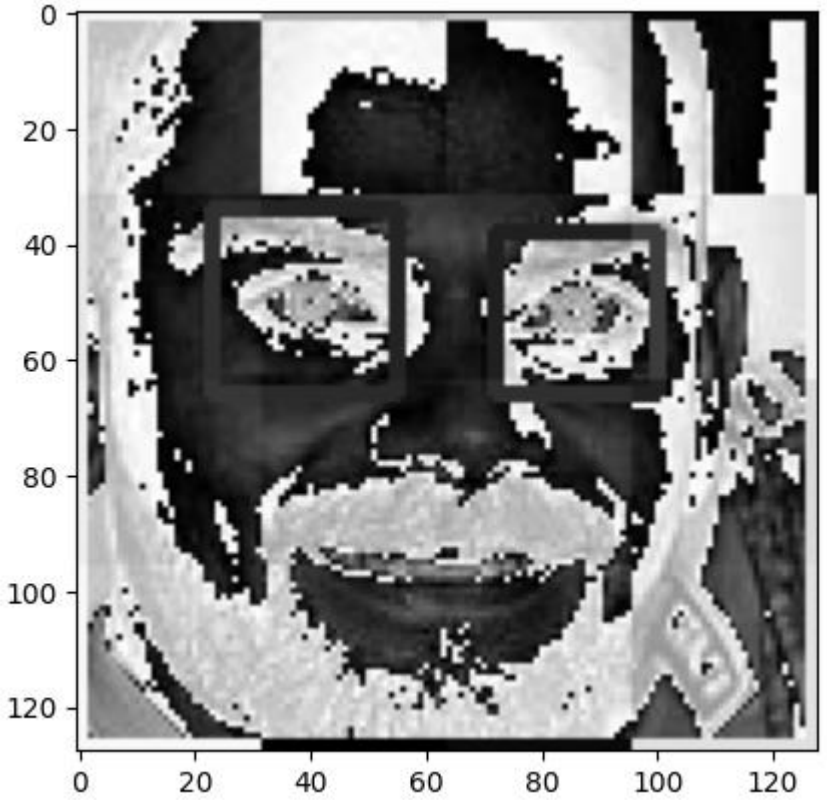
GrayScaling of Image

converting an image from color to grayscale.

Region of Interest



Differentiating all the facial features such as eyes nose and mouth etc.



Wavelet Transformation

Preprocessing

- Load image, detect face. If eyes >=2, then save and crop the face region

No eyes are matched from the previous images

Above image 2 eyes is None which means we should ignore this image and we will not use such image for model training

Data Cleaning

- **Made the folders automatically with some lines of code**
  - Created a script to automatically generate the necessary folder structure for organizing the image dataset into main directories and subdirectories for different categories, ensuring consistency and efficiency.
- **Manually examine cropped folder and delete any unwanted images**
  - Reviewed the automatically cropped images, identified and removed any misaligned or irrelevant images to maintain the quality of the dataset.

Grid Search

- GridSearch is a technique used in machine learning for hyperparameter tuning.
- It helps in finding the best combination of hyperparameters for a given model by exhaustively searching over a specified parameter grid.
- This ensures that the model performs optimally by selecting the set of hyperparameters that results in the highest performance metric on the validation set.

Outcomes for Gridsearch

```
best_estimators['svm'].score(X_test,y_test)
```

```
0.6885245901639344
```

```
best_estimators['random_forest'].score(X_test,y_test)
```

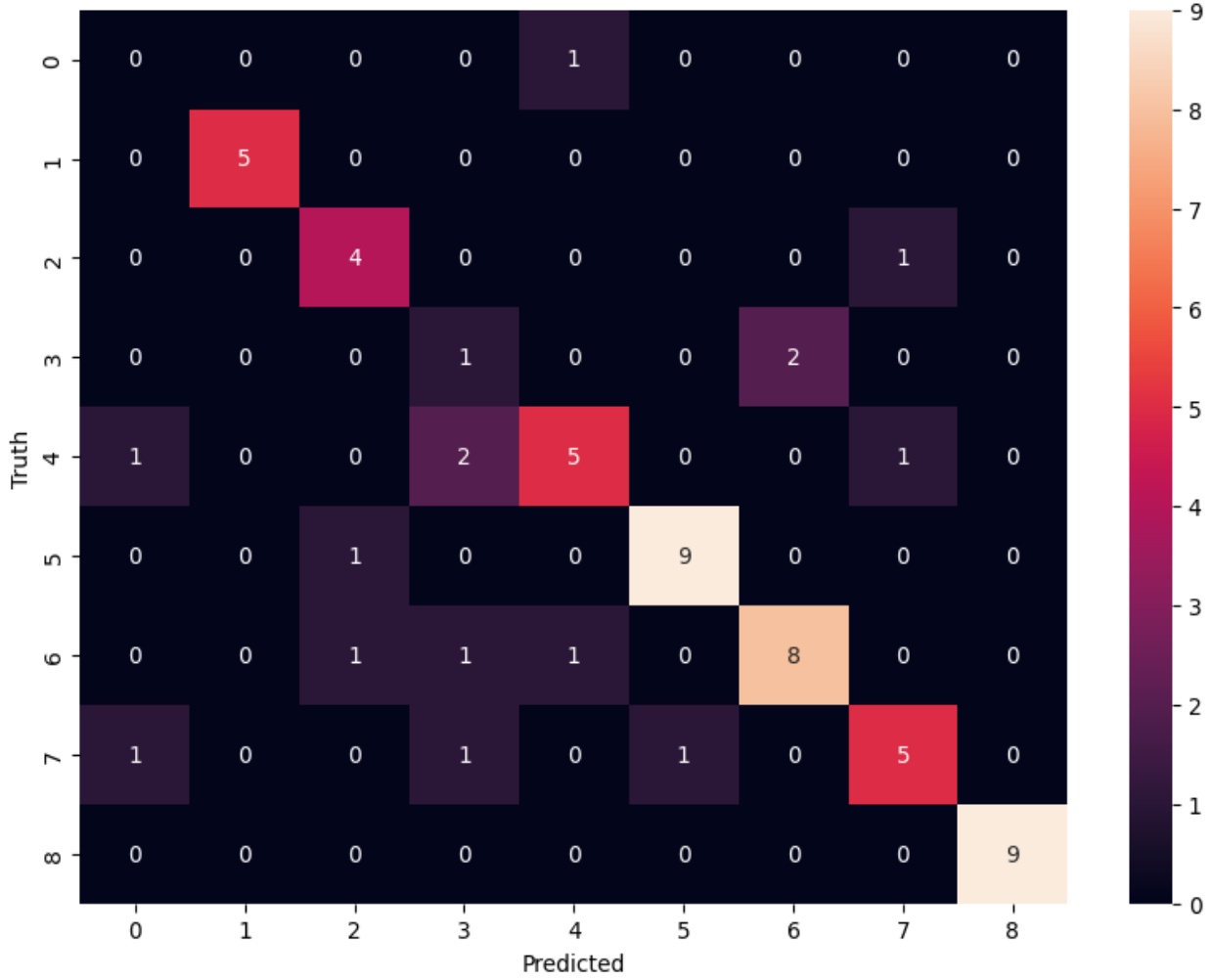
```
0.49180327868852456
```

```
best_estimators['logistic_regression'].score(X_test,y_test)
```

```
0.7540983606557377
```

Confusion Matrix

Serena Williams (0):
•Misclassified all instances. Not a single correct prediction.
•1 misclassified as Cristiano Ronaldo (7).
Virat Kohli (1):
•**5 correct predictions**.
•No misclassifications.
•**Lionel Messi (2)**:
•**4 correct predictions**.
•1 misclassified as Cristiano Ronaldo (7).
Kane Williamson (3):
•**1 correct prediction**.
•2 misclassified as Maria Sharapova (5).
Roger Federer (4):
•**5 correct predictions**.
•2 misclassified as Lionel Messi (2).
•1 misclassified as Serena Williams (0).
•1 misclassified as Maria Sharapova (5).
Maria Sharapova (5):
•**9 correct predictions**.
•1 misclassified as Kane Williamson (3).
Kobe Bryant (6):
•**8 correct predictions**.
•1 misclassified as Lionel Messi (2).
•1 misclassified as Kane Williamson (3).
Cristiano Ronaldo (7):
•**5 correct predictions**.
•1 misclassified as Serena Williams (0).
•1 misclassified as Lionel Messi (2).
•1 misclassified as Roger Federer (4).



Results

This is the b64 decoded image

```
"C:\Program Files\Python312\python.exe" N:\project\new\server\util.py
loading saved artifacts...start
loading saved artifacts...done
Class: serena_williams
Class Probability: [0.22, 0.67, 12.21, 0.1, 0.59, 0.14, 0.1, 45.19, 40.77]
Class Dictionary: {'Kane Williamson': 0, 'Kobe Bryant': 1, 'lionel_messi': 2, 'Maria Sharapova': 3, 'maria_sharapova': 4, 'roger_federer': 5, 'Ronaldo': 6, 'serena_williams': 7
```

```
, 'serena_williams': 7, 'virat_kohli': 8}
```

THANK YOU