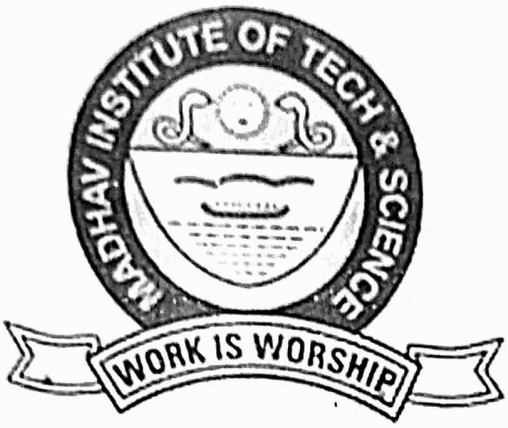
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade



Minor Project Report

on

Youtuber Classifier

Submitted By:

Ajay Parmar

0901AI211005

Faculty Mentor:

Dr. Kritika Bansal

Assistant Professor

CENTRE FOR ARTIFICIAL INTELLIGENCE
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Ajay Parmar** (0901AI211005) has submitted the project report titled **Youtuber Classifier** under the mentorship of **Dr. Kritika Bansal**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **Artificial Intelligence and Robotics** from Madhav Institute of Technology and Science, Gwalior.

23/11/23

Dr. Kritika Bansal
Faculty Mentor
Assistant Professor
Centre for Artificial Intelligence

23/11/23

Dr. R. R. Singh
Coordinator
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in **Artificial Intelligence & Robotics** at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Dr. Kritika Bansal**, Assistant Professor, Artificial Intelligence & Robotics

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Ajay Parmar
0901AI211005
III Year,
Centre for Artificial Intelligence

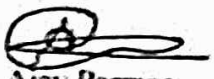
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR
(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)
NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence**, for allowing me to explore this project. I humbly thank **Dr. R. R. Singh**, Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Dr. Kritika Bansal**, Assistant Professor, Artificial Intelligence & Robotics, for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.



Ajay Parmar
0901AI211005
III Year,
Centre for Artificial Intelligence

ABSTRACT

The Youtuber Classifier utilizing image data involves a machine learning model designed to identify and categorize YouTubers based on their visual characteristics. The classifier operates on image datasets, extracting features and patterns to make predictions about the respective YouTuber category.

This innovative approach combines computer vision techniques with machine learning algorithms to discern distinguishing traits among YouTubers, potentially considering elements like facial recognition, channel branding, content types, and more. The model aims to automate the process of classifying YouTubers, providing insights into the diverse landscape of content creators on the platform.

TABLE OF CONTENTS

Chapter 1: Project Overview

1.1 Introduction

In the era of digital media, YouTube has become a massive platform hosting a diverse range of content creators. With millions of videos uploaded daily, it can be challenging to navigate through the vast content landscape. The "YouTube Content Creator Classifier" project aims to develop a machine learning model that automatically categorizes and classifies YouTubers based on their Images

1.2 Objective & Scope

1.2.1 Objective:

The objective of the "YouTubers Classifier Project" is to develop a machine learning model that can accurately classify and categorize YouTubers based on the visual content of their images, specifically focusing on their profile pictures, video thumbnails, or any other relevant visual elements.

1.2.2 Scope:

The scope of the "YouTubers Classifier Project" encompasses various aspects, including data collection, model development, evaluation, and potential deployment. Key elements of the project scope include:

Data Collection:

Gather a diverse dataset of YouTuber images, including profile pictures and video thumbnails.
Annotate the dataset with predefined categories or genres to facilitate supervised learning.
Preprocessing:

Extract relevant visual features from images, considering color, composition, and other characteristics.
Implement techniques to handle variations in image quality, resolution, and style.
Model Development:

Choose appropriate machine learning algorithms, such as Support Vector Machine (SVM) and Logistic Regression and Random forest Classifier, for image classification.
Train the model using the annotated dataset to learn patterns and associations between visual features and YouTubers.
Evaluation:

Assess the model's performance using metrics like accuracy, precision, recall, and F1 score. Conduct thorough testing to ensure the model generalizes well to new and unseen YouTuber images.
Integration with Recommendation System:

Integrate the YouTubers classifier into a broader recommendation system that considers user preferences and viewing history.
Implement mechanisms to update the model periodically to adapt to changing trends.

1.3 System requirements:

- Web Browser:- Compatibility with modern web browsers such as Chrome, Firefox, and Safari.
- Internet Connection:- A stable internet connection for real-time communication.
- Device Compatibility:- Support for various devices, including desktops, laptops, tablets, and smartphones.
- Python
- Open CV

Chapter 2: Literature & Review

2.1 Python

Python is a high-level programming language known for its readability and versatility. It's favored for its simplicity and ease of use, making it a go-to language for various applications, including web development, data analysis, artificial intelligence, and more. With a rich ecosystem of libraries and frameworks, Python simplifies complex tasks and encourages clean, efficient code. Its syntax emphasizes readability, which means that even newcomers find it accessible and understandable. Overall, Python's popularity stems from its combination of power, simplicity, and extensive community support.

2.2 Machine Learning

Machine learning is a subset of artificial intelligence that enables computers to learn and make predictions or decisions without explicit programming. It involves creating algorithms and models that can learn from data, identify patterns, and make predictions or take actions based on that learned information. Machine learning can be categorized into supervised learning, unsupervised learning, and reinforcement learning.

2.3 Key Features of Python & Machine Learning

Readability:

Python emphasizes code readability with a clear and expressive syntax. This makes it easy for developers to write and maintain code.

Extensive Standard Library:

Python comes with a large standard library that includes modules and packages for various tasks, reducing the need for external libraries.

Versatility:

Python is a versatile language that supports multiple programming paradigms, including procedural, object-orient

Chapter 3: Code & Result

3.1 Code

```
In [1]: import numpy as np
        import cv2
        import matplotlib
        from matplotlib import pyplot as plt
        %matplotlib inline

In [2]: img = cv2.imread('./test_images/sharapova1.jpg')
        img.shape

Out[2]: (555, 700, 3)

In [3]: plt.imshow(img)

Out[3]: <matplotlib.image.AxesImage at 0x12fa14918>




In [4]: gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        gray.shape

Out[4]: (555, 700)

In [5]: gray

Out[5]: array([[175, 175, 175, ..., 176, 175, 174],
               [175, 175, 175, ..., 177, 175, 174],
               [175, 175, 175, ..., 177, 176, 174],
               ...,
               [ 84,  87,  88, ..., 113, 113, 113],
               [ 88,  89,  90, ..., 113, 113, 113],
               [ 93,  91,  91, ..., 112, 112, 112]], dtype=uint8)

In [6]: plt.imshow(gray, cmap='gray')

Out[6]: <matplotlib.image.AxesImage at 0x12facb890>
```

```
In [19]: cv2.destroyAllWindows()
         for (x,y,w,h) in faces:
             face_img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
             roi_gray = gray[y:y+h, x:x+w]
             roi_color = face_img[y:y+h, x:x+w]
             eyes = eye_cascade.detectMultiScale(roi_gray)
             for (ex,ey,ew,eh) in eyes:
                 cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)


         plt.figure()
         plt.imshow(face_img, cmap='gray')
         plt.show()
```

Above cropped image no 2 eyes is None which means we should ignore this image and we will not use such image for model training

```
In [20]: path_to_data = "./dataset/"
         path_to_cr_data = "./dataset/cropped/"

In [21]: import os
         img_dirs = []
         for entry in os.scandir(path_to_data):
             if entry.is_dir():
                 img_dirs.append(entry.path)

In [22]: img_dirs

Out[22]: ['./dataset/technical_guruji',
          './dataset/bhuvan_bam',
          './dataset/sandeep_maheshwari',
          './dataset/mr_beast',
          './dataset/cropped',
          './dataset/pewdie_pie']
```

Go through all images in dataset folder and create cropped images for them. There will be cropped folder inside dataset folder after you run this code

```
In [23]: import shutil
         if os.path.exists(path_to_cr_data):
             shutil.rmtree(path_to_cr_data)
         os.mkdir(path_to_cr_data)
```

In [24]:
```
cropped_image_dirs = []
youtubers_names_dict = {}
for img_dir in img_dirs:
    count = 1
    youtubers_name = img_dir.split('/')[-1]
    youtubers_names_dict[youtubers_name] = []
    for entry in os.scandir(img_dir):
        if entry.is_file() and entry.name.endswith('.jpg', '.jpeg', '.png'):
            roi_color = get_cropped_image_if_2_eyes(entry.path)
            if roi_color is not None:
                cropped_folder = path_to_cr_data + youtubers_name
                if not os.path.exists(cropped_folder):
                    os.makedirs(cropped_folder)
                    cropped_image_dirs.append(cropped_folder)
                    print("Generating cropped images in folder: ", cropped_folder)
                cropped_file_name = youtubers_name + str(count) + ".png"
                cropped_file_path = os.path.join(cropped_folder, cropped_file_name)
                cv2.imwrite(cropped_file_path, roi_color)
                youtubers_names_dict[youtubers_name].append(cropped_file_path)
                count += 1
```

```
Generating cropped images in folder: ./dataset/cropped/technical_guruji
Generating cropped images in folder: ./dataset/cropped/bhuvan_bam
Generating cropped images in folder: ./dataset/cropped/sandeep_maheshwari
Generating cropped images in folder: ./dataset/cropped/mr_beast
Generating cropped images in folder: ./dataset/cropped/prashle_pie
```

Now you should have cropped folder under datasets folder that contains cropped images

Manually examine cropped folder and delete any unwanted images

Images in cropped folder can be used for model training. We will use these *raw* images along with wavelet transformed images to train our classifier. Let's prepare X and y now

In [27]:
```
X, y = [], []
for youtubers_name, training_files in youtubers_names_dict.items():
    for training_image in training_files:
        img = cv2.imread(training_image)
        scalled_raw_img = cv2.resize(img, (32, 32))
        img_har = w2d(img,'db1',5)
        scalled_img_har = cv2.resize(img_har, (32, 32))
        combined_img = np.vstack((scalled_raw_img.reshape(32*32*3,1),scalled_img_har.reshape(32*32,1)))
        X.append(combined_img)
        y.append(class_dict[youtubers_name])
```

In [28]:
```
len(X[0])
```

Out[28]: 4096

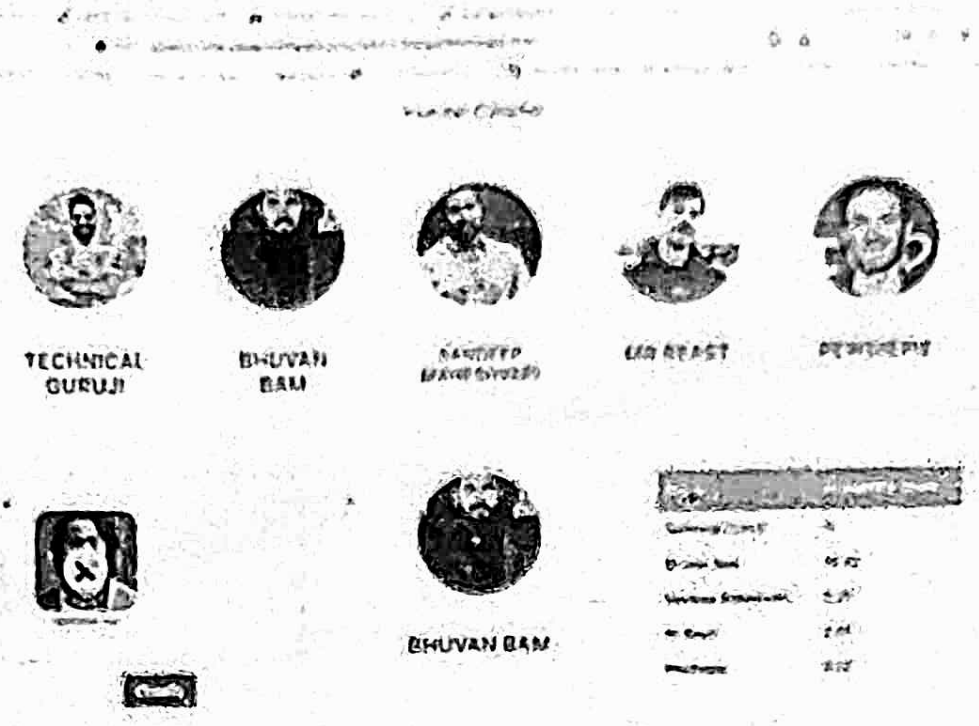
In [29]:
```
32*32*3 + 32*32
```

Out[29]: 4096

In [30]:
```
X[0]
```

Out[30]:
```
array([[ 73],
       [ 87],
       [ 89],
       ...,
       [239],
       [229],
       [  0]], dtype=uint8)
```

3.2 Result



3.3 Conclusion

Our trained model is working with 80 percent accuracy & hence it is able to detect the images of Youtuber accordingly by given datasets.