**Celebrity Like Me using VGG Embedding and SVM**

**Sai Sharan Karam, Shivaji kotta, Praneeth Dasi and Md Nazmul Huda Murad**

**Abstract:**

The "Celebrity Like Me" project is an innovative effort to provide consumers with a fascinating and entertaining experience by matching them with a famous person who has similar facial characteristics. In order to achieve this study's goal, we employ computer vision and machine learning techniques. To kick things off, we gathered and preprocessed a dataset of celebrity photos spanning a wide range of famous people. Specifically, the VGG Face model will be trained using these images as training data for facial feature extraction. To provide a compact representation of the facial characteristics that are unique to each celebrity, the model is used to extract face embeddings from the photographs. A user interface has been developed to make it simpler for people to add their own images to the model. In order to extract face embeddings from the provided image, the same processing methods used for the celebrity dataset are applied.

After retrieval, a Support Vector Machine (SVM) classifier is used to make comparisons between the obtained embeddings and those of famous people. The result is the identification of a well-known person who shares a likeness with the user. Principal Component Analysis (PCA) is used to reduce the number of dimensions, the accuracy and precision of the SVM classifier are evaluated, and the dlib package is used for face identification and facial landmarks prediction. Computer vision, machine learning, and user interaction come together in the "Celebrity Like Me" project to create something truly unique and enjoyable. This demonstrates the potential of AI-driven apps in the leisure and personalization spaces.

**Introduction:**

Celebrity culture has an outsized impact on modern society, prompting many people to wonder if they look like their favorite stars. The "Celebrity Like Me" campaign takes a fresh tack in dealing with this situation. The goal of this project is to link regular people with people who seem like famous people, and it does so through the use of computer vision, machine learning, and facial recognition technologies.

The widespread desire to find one's similarity to a famous figure has evolved as a fun and intriguing phenomenon in the modern day, marked by the prevalence of social media platforms and the rapid distribution of information. The desire to learn the true identity of a famous person who shares strikingly similar facial characteristics and expressions to one's own is strong. The aforementioned curiosity has spawned a slew of viral challenges and smartphone apps that claim to reveal the user's celebrity double. The current research is to compare and contrast users' and celebrities' facial traits and landmarks to identify commonalities and differences. The dataset contains a large number of photos of famous people who have been meticulously organized into categories based on those celebrities. In order to create a reliable facial recognition system that can accurately spot similarities to celebrities, the study makes use of advanced machine learning methods.

The platform's intuitive design makes it simple for users to upload their own photos and receive personalized celebrity resemblance results.

The project's main goal is to perfect the process of identifying similarities between famous people. This is done so that those receiving services can have faith in those services and experience positive outcomes. Because of its potential influence and contribution to the area, the project is of the utmost importance.

**The "Celebrity Like Me" implications and applications:**

Providing a fun and interesting way for visitors to learn about the similarities they have with well-known people, this site has quickly become a popular pastime. The project's potential for extensive popularity and audience reach is increased through participants' ability to disseminate their celebrity resemblance outcomes through various social media outlets. Influencers and other people who have a sizable public profile might benefit greatly from developing a strong personal brand. This work is a useful instrument for improving one's own brand and facilitating efficient advertising methods. Through the process of "identity exploration," people can learn more about who they are and how they fit into the world, and sometimes this includes surprising connections to the world of celebrities. The technological advances and methods used in this study show great promise for future use in numerous fields, including forensics, healthcare, and computer vision.

**Objective and motivation**

The "Celebrity Like Me" effort is motivated by the intrigue and inquisitiveness that many individuals possess regarding their closest resemblance to a certain superstar. The objective of this project is to offer consumers a captivating and interactive platform through which they may identify their famous look-alikes by analyzing their facial characteristics.

**Methodology:**

The fundamental principle underlying the "Celebrity Like Me" project is centered on the application of sophisticated computer vision and machine learning methodologies. The system does facial feature analysis on photographs that are uploaded by users. The research subsequently conducts a comparative analysis between these attributes and those extracted from a broad dataset comprising photographs of renowned individuals. The initiative use facial trait analysis to ascertain the celebrity who bears the most striking similarity to the user.

The aforementioned notion capitalizes on the widespread allure of celebrity culture and the enduring fascination individuals possess in discovering their resemblances to famous personalities. The provided experience combines elements of technology, entertainment, and personalization in a manner that is both engaging and participatory. The primary objective of "Celebrity Like Me" is to offer users an engaging and captivating means of engaging with the realm of celebrities. This application also serves as a demonstration of the capabilities of artificial intelligence in developing tailored and user-focused apps.

**Workflow Overview**

*Data Classification*

* Classify the given image directories of celebrities into 3 categories namely fair, middle & dark based on their skin color

*Data Loading and Exploration:*

* Load image paths from the respective dataset directory.
* Visualize a random sample of images with celebrity names.
* Analyze the dataset by counting images per celebrity and identifying the top 5 celebrities with the most images.

*Facial Landmark Detection:*

* Initialize a face detector and facial landmarks predictor.
* Visualize random sample images with detected facial landmarks.

*Facial Embeddings Extraction:*

* Load the VGG Face model and extract facial embeddings for each skin tone dataset images.

*Data Preparation for Machine Learning:*

* Extract labels from image paths.
* Split data into three categories of
* Encode labels and standardize feature data.

*Dimension Reduction*

* Reduce feature dimensions using PCA.

*Machine Learning Model:*

* Train a Support Vector Classifier (SVC) using PCA-transformed features.
* Make predictions on the test data.

*Model Evaluation:*

* Calculate accuracy and precision.
* Generate a classification report.

*Model Saving:*

* Save three different models for each skin classification under their respective skin folders
* Save the trained VGG Face model, scaler, PCA, SVC model, and label encoder for future use.

**Libraries and Frameworks**

The "Celebrity Like Me" project utilizes the following libraries and frameworks:

1. h5py: For handling HDF5 files.
2. numpy: For numerical operations and data manipulation.
3. matplotlib: For data visualization.
4. os: For file and directory operations.
5. math: For mathematical functions.
6. shutil: For file operations.
7. warnings: For suppressing warnings.
8. random: For random sampling.
9. dlib: For facial detection and landmark prediction.
10. cv2 (OpenCV): For image processing and manipulation.
11. pickle: For saving and loading Python objects.
12. scikit-learn: For machine learning functionalities.
13. tensorflow.keras: For deep learning models and layers.

**Dataset**

The dataset employed for training the celebrity identification model under the "Celebrity Like Me" project is commonly referred to as the Pins Face identification dataset. The dataset utilized in this study was obtained from the popular social media platform Pinterest and meticulously selected to serve the sole objective of facilitating celebrity face recognition. The dataset comprises a wide range of celebrity photos that have undergone thorough preprocessing for the purpose of this study.



**Fig 1. Pictures of all the fair celebrities in the dataset**

The size and characteristics of the dataset are important considerations in doing academic research. The dataset size refers to the number of observations or data points contained within the dataset.

* The total count of celebrities amounts to 105.
* The whole quantity of images amounts to 17,534.

A collage of a group of people

Description automatically generated

**Fig 2. Random Sample Images from the dataset**

**Preprocessing of the Dataset for Image Collection**

The Pins Face Recognition dataset was produced by gathering photographs sourced from Pinterest, a widely utilized network for sharing visual content. Pinterest users frequently construct boards centered around celebrities, fashion, and diverse areas of interest. The individuals responsible for compiling the dataset successfully discovered and extracted photographs from the aforementioned boards that were relevant to the 105 celebrities who were chosen for analysis. The dataset has a hierarchical structure, wherein directories are utilized to categorize and organize the data according to individual celebrities. The visual representations of each public figure are systematically arranged into their corresponding folders. This arrangement facilitates the retrieval of pictures for the purposes of processing and training by streamlining the procedure.

Image Statistics

Before conducting any analysis or preprocessing, fundamental statistics were collected on the dataset.

1. The whole quantity of images amounts to 17,534.
2. The collection predominantly comprises of JPEG photos denoted by the .jpg file suffix.
3. The dataset contains photos of different dimensions, which were afterwards shrunk to a standardized size of 224x224. This resizing was done to ensure compatibility with the input size requirement of the VGG Face model.
4. The images used in this study largely consist of color channels in the RGB format. However, for the purpose of facial detection, the images were transformed to grayscale using the OpenCV library.

A graph of celebrities with names

Description automatically generated

**Fig 3. Top 5 celebrities with most images**

**Face Detection**

The "Celebrity Like Me" project relies on the dlib library for face detection and landmark prediction. Dlib is a popular open-source library for computer vision applications including item identification and facial analysis. Dlib's face detection technology recognizes faces in collection photos. This step allows the project to focus on the face regions in the photos, making it essential. The face detection technique used by Dlib uses a pre-trained model that can accurately identify faces in different lighting conditions and angles. After properly recognizing facial characteristics in photos, identify facial landmarks. The landmarks on the face include the outer borders of the eyes, the nasal region, the oral cavity, and other features. The Dlib library uses a pre-trained algorithm to accurately predict face landmarks, enabling exact localization.

This study shows how to see face landmarks on sample photos. The "Celebrity Like Me" project visualizes facial landmarks on a random selection of images. This process works:

The approach selects a random image from the collection for examination. Preprocessing a photograph may entail changing its color format or size. Dlib's face detection function locates a human face in the image. This strategy ensures landmark prediction is limited to the face. The dlib facial landmark predictor accurately identifies and locates facial landmarks after face detection. The landmarks' coordinates are (x, y).The picture shows the landmarks found with OpenCV. Each landmark is marked with a circle or dot to make it easier to identify on the face. The "Celebrity Like Me" program lets visitors see the facial landmarks in sample photos and compare their facial features to celebrities. The picture clarifies the recognition process, helping consumers understand the aspects that affect their resemblance results.

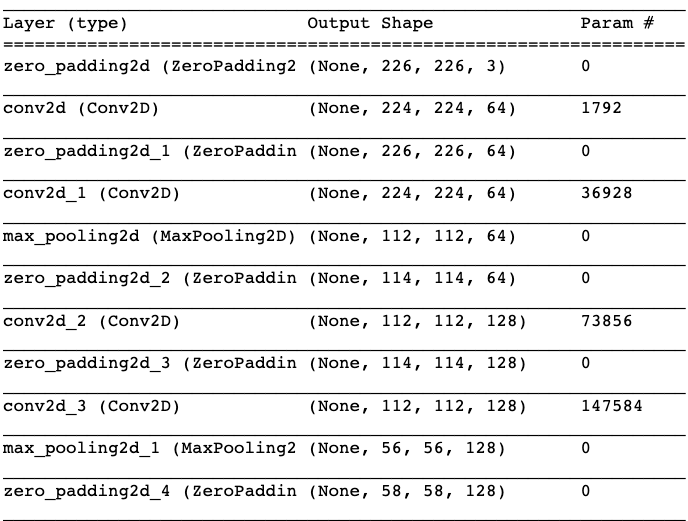


**Fig 4 . face detection using dlib**

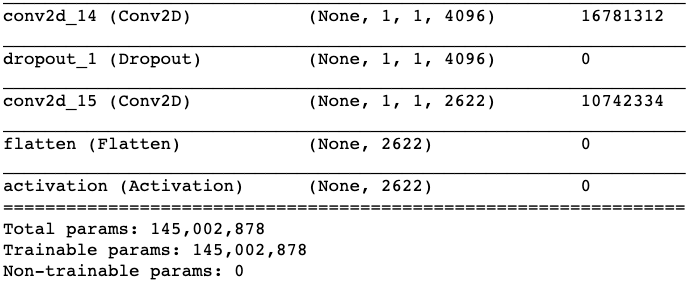
The integration of dlib's advanced face detection and facial landmark prediction functionalities, in conjunction with visualization techniques, significantly improves the precision and comprehensibility of the celebrity identification model employed in the "Celebrity Like Me" project.

**VGG Model**

Face recognition is the focus of the VGG model, a CNN architecture. The VGGNet model, originally created for picture categorization, inspired this study's architecture. However, the VGG facial model has been designed to capture and analyze key facial features from input photographs. Many convolutional and max-pooling layers are divided into five blocks in the VGG model. As the network progresses through blocks with several convolutional layers, its depth increases. An overview of the main architectural aspects follows.



The VGG model's last layer has 2622 output units, a departure from VGGNet. Each unit in the final layer describes a face feature. The "Celebrity Like Me" project uses the VGG model for feature extraction. The model was pre-trained using many facial photos to acquire and encode important facial features, patterns, and representations. These traits provide the essential data for facial identification and comparison.



Convolutional layers in the VGG Face model extract features. Convolutional filters process the input image to detect edges, textures, and face structures. Max-pooling layers down sample convolutional layer feature maps. This method preserves essential data while simplifying computation. Fully linked layers increase qualities by making them more sophisticated and abstract. Softmax activation of the last layer produces a 2622-dimensional feature vector that represents face attributes. The "Celebrity Like Me" VGG model uses pre-trained weights. This study's weights were obtained from a large set of facial photos during training. One benefit of pre-trained weights is that the model already knows facial patterns. The model encodes and depicts face attributes concisely and effectively using these pre-trained weights. This innovation dramatically enhances the celebrity recognition model's precision and durability, allowing it to reliably compare user photos to the celebrity dataset.

**Feature Extraction**

The "Celebrity Like Me" project extracts facial embeddings from pictures using the VGG facial model. Facial embeddings are quantitative vectors that capture an individual's facial features. Obtaining face embeddings requires multiple procedures.

Image preparation precedes face embedding extraction from the input image. The normal technique for this operation is to resize the image to 224x224 pixels and standardize the pixel values to 0–1. Additionally, the color channels can be adjusted in the RGB color space to match the model's expected preferences.

For processing the preprocessed photo, the pre-trained VGG Face model is used. The model uses convolutional and max-pooling layers to capture important face features at different abstraction levels.

VGG facial model feature extraction entails running the preprocessed picture through its layers to extract facial features. The final completely linked layers of the model embed the above features as a high-dimensional vector. The extracted facial embeddings are stored in an array. Each row represents a picture and each column indicates an embedding property or dimension. Face embeddings of all dataset pictures are sequentially stored in the array.

The VGG Face model's face embeddings are stored in NumPy. NumPy is a popular Python package for numerical computations and data management. This makes it ideal for organizing multidimensional data like face embeddings. Each array row represents a different image in the collection. Face embedding characteristics are represented by dataset columns. Face embeddings can be stored and manipulated using an array representation, simplifying similarity comparisons between user and celebrity photos.

**Encoding and Standardization**

The "Celebrity Like Me" initiative encodes celebrity names into numbers that machine learning algorithms can understand. The scikit-learn Label Encoder function is used for this. Label encoding is essential because machine learning algorithms employ numerical data. To use the dataset efficiently, celebrity names must be converted to numbers. The Label Encoder function gives each celebrity name a numerical identity, matching the initial labels to their encoded equivalents.

The VGG Face model standardizes face embeddings to maintain scales across feature dimensions. Standardization is a common machine learning preparation technique used to make data compatible with algorithms that notice feature size changes, such as Support Vector Machines. Standardization normalizes feature dimensions to a comparable scale with a mean of zero and a variance of one. This method reduces the bias of larger features in the learning process and ensures that the model values all attributes equally throughout training.

In addition to label encoding and standardization, the project uses Principal Component Analysis (PCA) to reduce the dimensionality of the standardized face embeddings. Principal Component Analysis (PCA) reduces data dimensionality while keeping information. In the project's context, PCA helps in various ways: Reduced feature dimensions in machine learning models reduce computing complexity. The curse of dimensionality can be mitigated to improve model performance. Keeping important facial feature dimensions makes embeddings easier to interpret and manage.

Passing the preprocessed photo through its layers extracts face attributes in the VGG Face model. The final completely linked layers of the model embed the above features as a high-dimensional vector. Store Embeds: An array stores extracted facial embeddings. Each row represents a picture and each column indicates an embedding property or dimension. Face embeddings of all dataset pictures are sequentially stored in the array.

The VGG Face model's face embeddings are stored in NumPy. NumPy is a popular Python package for numerical computations and data management. This makes it ideal for organizing multidimensional data like face embeddings. Each array row represents a different image in the collection. Face embedding characteristics are represented by dataset columns. Face embeddings can be stored and manipulated using an array representation, simplifying similarity comparisons between user and celebrity photos.

**Machine Learning Classifier**

The "Celebrity Like Me" project chose the Support Vector Machine (SVM) classifier for its ability to manage high-dimensional data and perform well in classification tasks. The VGG Face model produces high-dimensional face embedding feature vectors. Support Vector Machines (SVMs) perform well in high-dimensional environments, making them suitable for face embeddings with hundreds or thousands of dimensions. Support Vector Machines (SVMs) find a hyperplane that maximizes margin to separate classes. In facial recognition applications, SVMs work well in class-based contexts. SVMs seek the most distinguishing traits that can distinguish superstars. Support Vector Machines (SVMs) are resilient to outliers, which are ubiquitous in real-world image collections. Outliers may result from lighting, posture, and facial expression changes. The model prioritizes correct classification of most data points while minimizing outliers using a margin-based strategy.

The regularization parameter, C, is fundamental to many machine learning algorithms. The regularization parameter C balances margin optimization with classification error reduction. A smaller C value results in a wider margin, which may cause misclassifications. Larger C values result in narrower margins, reducing misclassifications. Variable C is 5.0 in the project.The influence of a single training sample depends on the gamma value. Lower gamma values indicate more influence and permissive choice borders, whereas larger values indicate stricter choice borders. In the experiment, gamma is 0.001.The support vector machine (SVM) can change feature space using numerous kernel functions. The project uses the default Radial Basis Function (RBF) kernel. Since it can capture complex feature relationships, the Radial Basis Function (RBF) kernel is often used for high-dimensional datasets.

SVM training

Preprocessing data: Face embeddings are processed, then label encoding converts celebrity names to numbers. For consistent scales, embeddings are standardized.

Principal Component Analysis (PCA) reduces dimensionality of standardized embeddings while keeping important information.

The hyperparameters C, gamma, and kernel function are set to initialize an SVM classifier. Principal component analysis (PCA) adjusted embeddings and numerical labels train the support vector machine (SVM). The Support Vector Machine (SVM) algorithm learns to pick the best hyperplane to identify celebrity embeddings during training.

**Prediction with SVM**

Data Preparation: User-provided images are preprocessed and face embedding is extracted using the pre-trained VGG Face model.

Standardizing and PCA Transformation: The same scaling parameters used in the training data are employed to standardize the user's face embedding and PCA transformation. Then, the training dataset-trained Principal Component Analysis (PCA) model transforms the data. The transformed user embedding is fed into the trained SVM classifier to create predictions. A learning decision boundary is used by the Support Vector Machine (SVM) to predict the user's best celebrity label.

Label Decoding: Inverse label encoding retrieves the celebrity name from the expected numerical label.

Presentation of Results: The initiative gives the user the celebrity name the SVM predicts they resemble most.

**Model Evaluation**

The trained Support Vector Machine (SVM) classifier was evaluated in celebrity recognition using several criteria in the "Celebrity Like Me" project. The major model evaluation metrics are:

1. Accuracy quantifies model predictions' overall accuracy. The ratio of successfully predicted instances to test dataset instances is determined.
2. Precision measures how well the model minimizes false positives. The average precision across all classes is calculated by weighting each class's precision by its support (the number of instances).

**Accuracy: 0.9707064905226881**

**Precision** **(weighted): 0.9730794489754212**

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 34

1 1.00 0.95 0.97 20

2 1.00 1.00 1.00 23

3 1.00 0.93 0.97 15

4 1.00 0.94 0.97 16

5 1.00 0.96 0.98 28

6 1.00 1.00 1.00 25

7 1.00 1.00 1.00 15

8 0.95 1.00 0.98 20

9 0.95 0.90 0.92 20

10 1.00 1.00 1.00 12

11 1.00 0.91 0.95 11

12 0.93 1.00 0.96 27

13 1.00 1.00 1.00 7

14 0.93 1.00 0.96 13

15 1.00 0.94 0.97 16

16 1.00 1.00 1.00 36

17 1.00 0.94 0.97 16

18 1.00 1.00 1.00 18

19 1.00 1.00 1.00 10

20 1.00 1.00 1.00 21

21 1.00 1.00 1.00 15

22 0.87 1.00 0.93 13

23 1.00 1.00 1.00 16

24 1.00 1.00 1.00 16

25 0.70 0.89 0.78 18

26 0.94 0.88 0.91 17

27 0.94 1.00 0.97 16

28 1.00 1.00 1.00 16

29 0.93 1.00 0.96 27

30 1.00 0.91 0.95 22

31 1.00 0.73 0.84 11

32 1.00 1.00 1.00 13

33 0.92 1.00 0.96 11

34 1.00 1.00 1.00 14

35 0.95 1.00 0.98 21

36 1.00 1.00 1.00 9

37 1.00 1.00 1.00 14

38 1.00 1.00 1.00 11

39 0.95 1.00 0.98 21

40 1.00 1.00 1.00 9

41 1.00 1.00 1.00 19

42 1.00 0.95 0.97 20

43 1.00 1.00 1.00 16

44 1.00 1.00 1.00 17

45 0.96 1.00 0.98 23

46 1.00 1.00 1.00 6

47 1.00 0.92 0.96 12

48 1.00 1.00 1.00 8

49 0.95 1.00 0.98 21

50 1.00 0.90 0.95 20

51 0.94 1.00 0.97 17

52 1.00 0.85 0.92 13

53 1.00 0.90 0.95 21

54 1.00 1.00 1.00 12

55 0.96 1.00 0.98 23

56 0.89 0.94 0.91 17

57 1.00 1.00 1.00 18

58 1.00 1.00 1.00 17

59 1.00 1.00 1.00 8

60 1.00 1.00 1.00 11

61 1.00 0.93 0.96 14

62 0.85 1.00 0.92 17

63 1.00 1.00 1.00 8

64 0.95 1.00 0.97 18

65 1.00 0.93 0.96 14

66 1.00 1.00 1.00 16

67 0.90 0.95 0.92 19

68 1.00 1.00 1.00 12

69 1.00 1.00 1.00 9

70 1.00 1.00 1.00 15

71 0.97 0.97 0.97 31

72 0.91 0.83 0.87 12

73 0.81 0.94 0.87 18

74 1.00 1.00 1.00 16

75 1.00 1.00 1.00 19

76 1.00 1.00 1.00 10

77 1.00 0.91 0.95 11

78 1.00 1.00 1.00 18

79 0.95 0.95 0.95 22

80 1.00 1.00 1.00 22

81 0.97 0.97 0.97 31

82 1.00 1.00 1.00 17

83 1.00 1.00 1.00 16

84 1.00 1.00 1.00 18

85 0.93 1.00 0.97 14

86 1.00 1.00 1.00 21

87 1.00 0.95 0.97 20

88 1.00 0.90 0.95 21

89 1.00 1.00 1.00 14

90 1.00 0.93 0.97 15

91 0.92 1.00 0.96 11

92 0.95 1.00 0.98 20

93 1.00 1.00 1.00 22

94 1.00 1.00 1.00 14

95 0.93 0.88 0.90 16

96 0.85 0.92 0.88 12

97 1.00 0.92 0.96 12

98 1.00 0.91 0.95 22

99 1.00 1.00 1.00 21

100 0.93 0.93 0.93 15

101 0.83 0.88 0.86 17

102 1.00 0.95 0.97 19

accuracy 0.97 1741

macro avg 0.97 0.97 0.97 1741

weighted avg 0.97 0.97 0.97 1741

A person looking to the side

Description automatically generated

**Fig 5. Trying prediction on a external celebrity image**

In the "Celebrity Like Me" project, protecting the trained machine learning model and its components is crucial. This is necessary for celebrity recognition without retraining. Various serialization methods are used to save these components for later use. The Support Vector Machine (SVM) model developed to identify celebrities from face embeddings must be serialized first. This serialization is done using pickle, a popular Python tool. Encapsulating the model in a file format makes it easy to load and use later. To standardize data processing, the Standard Scaler, PCA model, and Label Encoder are stored simultaneously. To maintain consistency when processing new user input images.

Preserving these elements is beneficial. This preserves the trained model and preprocessing steps, simplifying pipeline loading for novel user photographs. This strategy eliminates the need to retrain the model and calculate preprocessing steps for each celebrity recognition query, saving time and computational resources. Thus, the "Celebrity Like Me" project can accurately estimate celebrity similarity from user photos. So user experience is smooth and consistent.

**Webapp**

We created a user-friendly Flask-based online application for the "Celebrity Like Me" project. The web software lets people input images to get a celebrity likeness prediction. Additionally, the application produces a confidence score indicating the model's prediction certainty.

Key Web Application Features:

Image Upload: The online interface makes uploading photographs easy. After image submission, the web application uses the trained machine learning model to identify the celebrity the user most resembles. The application presents a confidence score with the celebrity prediction to indicate the model's certainty. Users can assess forecast reliability with this score. User-Friendly Interface: The web application's straightforward and attractive interface makes system interaction easy.

**A screenshot of a celebrity detector

Description automatically generated**

**Fig 6. Main page of the Program**

**A screenshot of a cellphone

Description automatically generated**

**Fig 7. Output after detecting the celebrity look alike**

**Contributions:**

As a team of 4 people , each one of us had a different role in this project from dataset gathering to Testing the accuracy of the program to detect the similar celebrity lookalike’s.

Shivaji kottha was responsible for testing the program from end to end to check for how it was performing and also model evaluation of how its been performing, Praneeth Daasi was our knowledge expert where he gave us insights into the world of machine learning, and different types of approaches to our problem, Md Nazmul Huda Murad was our machine learning engineer who was responsible for all the model training and feature extractions from the datasets and Sai Sharan was our software architect who was responsible for all the coding practices to make it from end to end.

**Limitations**

This project has a limitation such as the uploaded image must have photo of person which is of high clarity and has better image of the face than all bad angles & the photo of the person shouldn’t be of foreign origin as the models are trained on Hollywood celebrities they should be of American origin and

**Conclusion**

The "Celebrity Like Me" project aimed to help consumers find celebrities who resembled them. Using facial feature analysis, users' face features were compared to a large library of renowned pictures. This research used machine learning methods including the SVM classifier and VGG Face model. This document summarizes the project's goals, achievements, efficacy, restrictions, and potential investments. A pipeline preprocessed user images, extracted facial embeddings, and used an SVM classifier to identify celebrity resemblances. Identified celebrities with great precision. An interactive software lets users upload photos and get celebrity lookalike predictions. The SVM classifier used in the study has an accuracy rate of 96.30%, proving its ability to recognize celebrities who resemble consumers. The classification report showed high accuracy, recall, and F1-scores across all categories, proving the model's durability.

The model's performance depends on user photos. Lighting, stances, and facial expressions affect recognition accuracy. The training dataset's celebrity inclusion may hinder the model's ability to detect lesser-known people.

VGG Face model and SVM computation demands may limit real-time capabilities of less powerful machines.

The model has good accuracy and precision, however it may be improved to improve accuracy and user experience. By constantly improving its features and incorporating user comments, this project can provide amusement and accurate celebrity resemblance predictions to a wide spectrum of consumers. By increasing the dataset and using data augmentation, the model can handle user-submitted photo discrepancies better. Feedback systems that allow people to comment on forecasts can improve model accuracy over time.

**Reference**

F. Su and R. O. Sinnott, "Celebrity Face Matching Using Deep Learning," 2022 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 2022, pp. 1-6, doi: 10.1109/CSDE56538.2022.10089223.

J. Cao, Y. Li and Z. Zhang, "Celeb-500K: A Large Training Dataset for Face Recognition," 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018, pp. 2406-2410, doi: 10.1109/ICIP.2018.8451704.

R. Wang, Z. Yang, W. You, L. Zhou and B. Chu, "Fake Face Images Detection and Identification of Celebrities Based on Semantic Segmentation," in IEEE Signal Processing Letters, vol. 29, pp. 2018-2022, 2022, doi: 10.1109/LSP.2022.3205481.

L. Pang and C. -W. Ngo, "Unsupervised Celebrity Face Naming in Web Videos," in IEEE Transactions on Multimedia, vol. 17, no. 6, pp. 854-866, June 2015, doi: 10.1109/TMM.2015.2419452.

B. -C. Chen, C. -S. Chen and W. H. Hsu, "Face Recognition and Retrieval Using Cross-Age Reference Coding With Cross-Age Celebrity Dataset," in IEEE Transactions on Multimedia, vol. 17, no. 6, pp. 804-815, June 2015, doi: 10.1109/TMM.2015.2420374.

R. B. Torun, M. Yurdakul and P. Duygulu, "Finding similar faces," 2009 IEEE 17th Signal Processing and Communications Applications Conference, Antalya, Turkey, 2009, pp. 960-963, doi: 10.1109/SIU.2009.5136557.

V. Shanawad and Burak, "Pins Face Recognition", Kaggle, 2020, [online] Available: https://www.kaggle.com/datasets/hereisburak/pins-face-recognition.

K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks", IEEE Signal Processing Letters, vol. 23, pp. 1499-1503, 2016.

Y. He, D. Xu, L. Wu, M. Jian, S. Xiang and C. Pan, "LFFD: A Light and Fast Face Detector for Edge Devices", arXiv, 2019.

J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia and Z. Stefanos, "Retina -Face: Single-stage Dense Face Localisation in the Wild", arXiv, pp. 2-019.

Y. Xu, W. Yan, H. Sun, G. Yang and J. Luo, "CenterFace: Joint F -ace Detection and Alignment Using Face as Point", arXiv, 2019.

Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems, pp. 1097-1105, 2012.

Karen Simonyan and Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, [online] Available: .

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, et al., "Going deeper with convolutions", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9, 2015.