StarMirror: Celebrity Doppelgängers with EfficientNet

Anna Torell
University of Michigan
Ann Arbor, MI
atorell@umich.edu

Ryan Galligan
University of Michigan
Ann Arbor, MI
ryangall@umich.edu

Jack Seel
University of Michigan
Ann Arbor, MI
seeljack@umich.edu

1. Introduction

Our work stems from a common, lighthearted dilemma: the infamous ”Who’s your celebrity crush?” question, which often leads to awkward moments in relationships. StarMirror provides a fun solution by identifying the celebrity who most resembles your partner, offering a playful way to navigate the scenario. Beyond this, the project addresses a huge challenge in facial recognition: creating unbiased, accurate models for identifying and comparing faces.

Our contributions include leveraging the *EfficientNet V2* [4] framework to build a robust model that extracts key features including gender, smile, and age. By integrating data augmentation, custom layers, and normalization, we improved accuracy in facial recognition. The result is a website that generates celebrity doppelgangers from an image and can compare similarity percentages between two faces, enabling reproducibility and extensive testing. This engaging application has meaningful impacts in facial detection.

2. Related Work

There are many iterations of this type of project that can be seen in cellphones, airports, and more. Most use traditional deep learning concepts, but some use new AI technologies. Work related to facial recognition dates back to the 1990s. In *Finding Your Celebrity Look Alike* [2] they use the IMDB wiki dataset with gender and age labels. They used the VGG-Face architecture to find vectors and the Haar Cascade to detect faces. They used Euclidean distance to score the model. The strength of this paper is the use of preexisting models to create a robust model. A weakness of this model is the arbitrary scoring system that does not allow for true understanding of performance.

Another project, *Find a Celebrity Doppelganger Using Dlib* [3] uses Dlib RESNET to identify celebrities that are identical to each other. Celebrity images are converted to vectors. This process is applied to the test images, and the resulting vectors are compared using the Euclidean norm to evaluate them against a threshold. If the distance is below the threshold, there is a match. A strength of this model is the ability to compare images quickly because of dlib. A weakness of this approach is that there is little to measure how well the model is working. There is no real measure of accuracy.

To specifically recognize facial features, we are relying on a pre-defined CNN structure, like these other papers. We are utilizing transfer learning for efficiency. We will not be using vector-based comparisons but rather embeddings to be able to further fine tune. We will not be using only euclidean distance but also cosine similarity with top-k accuracy to output the top 5 celebrities. Accuracy was a concern in both of these works, so testing multiple options for a robust numerical output is the correct option for us.

3. Methods

In this project, we propose a celebrity image retrieval approach based on a deep learning model, specifically utilizing *EfficientNet V2* [4], an efficient convolutional neural network (CNN) pre-trained on ImageNet. EfficientNet V2 has been chosen for its ability to find a balance between model performance and computational efficiency. As we struggled to find RAM resources on Google Colab, we switched to Kaggle. The model is optionally unfrozen, to extract features from input images and fine-tune the model with the addition of our own layers. These features capture high-level visual information, such as textures, shapes, and identities, making them ideal for measuring visual similarity between images.

EfficientNetV2, a version of EfficientNet, is optimized for performance in terms of accuracy and computational cost. For image pre-processing, all images are resized to 224x224 pixels and normalized to the range [0, 1], ensuring compatibility. Then, images are augmented with flipping, rotating, and zooming to make the results more robust. The input image is processed by the feature extractor to produce a fixed-length embedding, which is then used for comparison with the embeddings of other images in the dataset. The specific layers are mentioned fine tuning hyperparameters section. To measure the similarity between images,

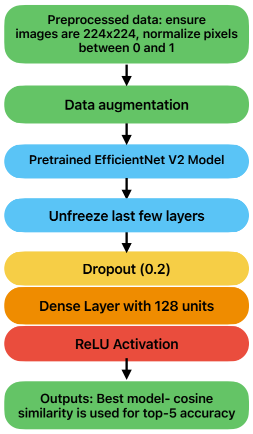
Preprocessed data: ensure images are 224x224, normalize pixels between 0 and 1
Data augmentation
Pretrained EfficientNet V2 Model
Unfreeze last few layers
Dropout (0.2)
Dense Layer with 128 units
ReLU Activation
Outputs: Best model- cosine similarity is used for top-5 accuracy

Figure 1. Model Architecture

both euclidean distance and cosine similarity are employed interchangeably to compare output differences with different accuracy measures. The top-5 most similar images are identified based on their cosine similarity scores.

4. Experiments

The data set used is the *Celebrity-1000* [1] dataset from Hugging Face, which contains 1000 unique celebrities with around 18 images per celebrity. Originally, we wanted to use a set with more celebrities for increased diversity, but did not have the resources.

In addition to the pre-processing hyperparameters already discussed, our fine-tuning includes a dropout rate of 0.2, a dense layer with 128 units, and a ReLU activation function. We keep pre-trained layers frozen and unfreeze the last few custom layers. We did a lot of experimentation with layers but the addition of more layers and more customization made our model overfit to small details such as glasses, background, and clothing.

We tested both euclidean distance and cosine similarity for evaluation, cosine similarity performed much better so our final model utilizes that. Cosine similarity is calculated between the embeddings of a given input image and the embeddings of all images in the dataset. The similarity score was reported as a percentage, which corresponds to the cosine similarity multiplied by 100. For top-k retrieval, for each input image, the top 5 most similar images were retrieved, and their similarity scores were displayed alongside their corresponding celebrity names.

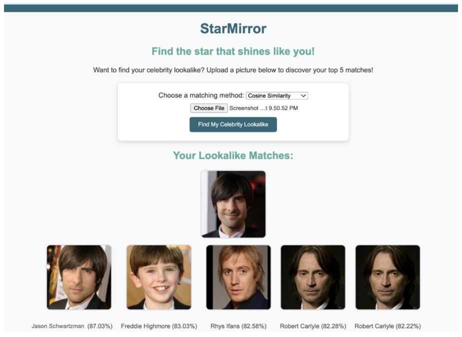
StarMirror
Find the star that shines like you!
Want to find your celebrity lookalike? Upload a picture below to discover your top 5 matches!
Choose a matching method: Cosine Similarity
Choose File Screenshot …t 9.50.52 PM
Find My Celebrity Lookalike
Your Lookalike Matches:
Jason Schwartzman (87.03%)
Freddie Highmore (83.03%)
Rhys Ifans (82.58%)
Robert Carlyle (82.28%)
Robert Carlyle (82.22%)

Figure 2. Baseline Result

The baseline approach used for this unique challenge was inputting unseen images of celebrities that were already in the dataset. With cosine similarity, these celebrities matched with themselves with a match rate or 85% or above based on cosine similarity. Euclidean distance performed worse in this case. This baseline allowed us to tune our model.

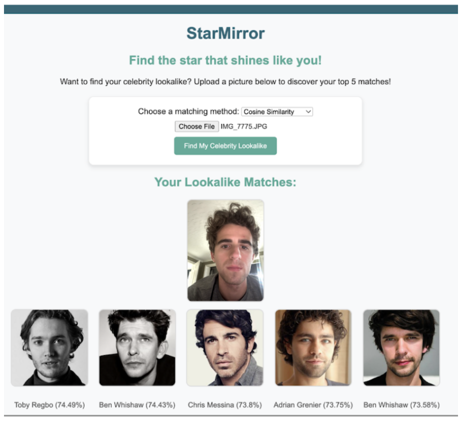
StarMirror
Find the star that shines like you!
Want to find your celebrity lookalike? Upload a picture below to discover your top 5 matches!
Choose a matching method: Cosine Similarity
Choose File IMG_7775.JPG
Find My Celebrity Lookalike
Your Lookalike Matches:
Toby Regbo (74.49%)
Ben Whishaw (74.43%)
Chris Messina (73.8%)
Adrian Grenier (73.75%)
Ben Whishaw (73.58%)

Figure 3. Test Results

This is a unique case for neural networks where a train and test set does not apply. In this case, the baseline should be higher than test results with random images. The test images, around 50 images of diverse friends and online photos, all have a match of above 50% with a celebrity that resembles them based on cosine similarity.

5. Conclusion

StarMirror showcases a creative use of facial recognition techniques in a fun context, leveraging the EfficientNet V2 model to identify celebrity lookalikes with high accuracy using cosine similarity. Our approach outperformed methods like Euclidean distance, balancing both efficiency and performance. Reproducibility is enabled by available datasets, pre-trained architectures, and the website implementation, allows others to replicate or extend the work. Despite the success, the model is sensitive to details like accessories or backgrounds, which can impact accuracy. Future improvements could involve larger, more diverse datasets, or other intricate architectures. Expanding the scope to include emotion recognition or context-aware matching could unlock different, real-world applications. This project highlights a balance between leveraging existing methods and creating user-focused application with potential for development.

References

[1] Tony Assi. Celebrity-1000 dataset. *Hugging Face Datasets*, 2024. Accessed: 2024-12-10. 2

[2] Davis Chase and Amanda Jacquez. Finding Your Celebrity Look Alike. pages 1–5, 2019. 1

[3] Poulastya Mukherjee. Find a Celebrity Doppelganger Using Dlib. 2021. 1

[4] Mingxing Tan Tan and V. Quoc Le. Efficientnetv2: Smaller models and faster training. *Cornell University*, pages 1–11, 2021. 1