

Face Recognition of Images Obtained from the Internet (TPA ID:2)

Siddhartha Guggulotu, *ME20B167*, and Jnaneswara Rao Rompilli, *EE20B052*,

Abstract

In this Face Recognition (FR) project, we have used a small set of labeled data of the current IITM faculty along with a mix of few standard datasets. We have fine tuned a pre-trained model called Arcface. After the training and validation steps, the model's performance is tested using new data. What makes this project unique is the real-world application—during testing, the software needs to go through web pages, find face images, and recognize them. This mirrors situations where FR is used on websites, highlighting the practical significance of effective face recognition in online content.

I. INTRODUCTION

THE primary objective of this project is to implement a robust system capable of identifying individuals from images sourced from the internet. The envisioned solution involves the automatic extraction of facial images from a given webpage, followed by the application of a sophisticated face recognition algorithm. The algorithm assigns name labels to recognized individuals, leveraging a trained model on diverse datasets encompassing Indian Cricketers, International Cricketers, Bollywood celebrities, and the esteemed faculty members of the IIT Madras Computer Science Department.

A. Dataset Compilation

[link to dataset zip file](#)

Our model's foundation rests on a meticulously curated dataset, a fusion of three Kaggle datasets and a collection crafted from Google Images featuring the distinguished faculty at IIT Madras.

The dataset is categorized as follows:

- **Indian Cricketer's Images:** 15 individuals, with an average of 30-40 images per person.
- **Faculty:** 45 esteemed members from IIT Madras, with a range of 3-10 images per person.
- **Cricket Players Faces:** A diverse set of 70 individuals, with an average of 10-20 images per person.
- **Indian Actor Images Dataset:** A substantial collection of 6750 images, with 40-45 images per person, showcasing the richness and variety of Indian cinema.

II. ALGORITHMIC DESCRIPTION

A. Face Recognition Framework

1) **Web Scraper:** To gather a diverse dataset, we developed a sophisticated web scraper capable of extracting images with faces from both HTTP and HTTPS websites. Leveraging Python libraries such as BeautifulSoup for parsing and Face Recognition for face detection, our scraper ensures the inclusion of high-quality images with facial features.

2) **Siamese Network:** A Siamese network is a neural network architecture used in computer vision for similarity or dissimilarity comparison tasks. The key components and working principle are as follows:

Architecture:

- **Twin Networks:** Two identical neural networks, referred to as the "twin" networks, share the same architecture.
- **Siamese Structure:** The twin networks process pairs of input images simultaneously, which can be similar or dissimilar.
- **Feature Extraction:** Both networks employ convolutional layers to extract relevant features from the input images.
- **Embedding Layer:** The extracted features go through an embedding layer, producing fixed-size vectors (embeddings) representing essential features.
- **Distance Metric:** A layer calculates the similarity or dissimilarity between the embeddings using a distance metric (e.g., Euclidean distance or cosine similarity).

Training Process:

- **Pair Generation:** During training, pairs of similar or dissimilar instances are generated.
- **Loss Function:** The network is trained using a loss function that minimizes the distance between similar pairs and maximizes the distance between dissimilar pairs.
- **Backpropagation:** Gradients are backpropagated through both branches, updating shared parameters to improve the model's discrimination capability.

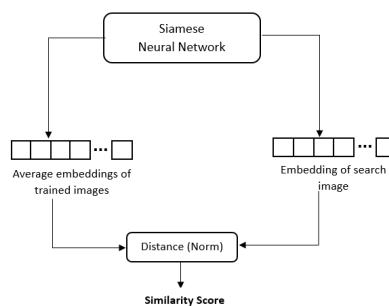


Fig. 1. Siamese network

3) **ArcFace**: Our selected face recognition algorithm, ArcFace, stands out for its precision and resilience. In contrast to conventional approaches, ArcFace incorporates angular margin-based metric learning, elevating the discriminative capabilities of feature embeddings. This not only refines recognition accuracy but also empowers the model to adeptly manage pose and illumination variations. Furthermore, for our project, we leverage a pretrained version of the ArcFace model, streamlining the integration of its sophisticated capabilities into our framework.

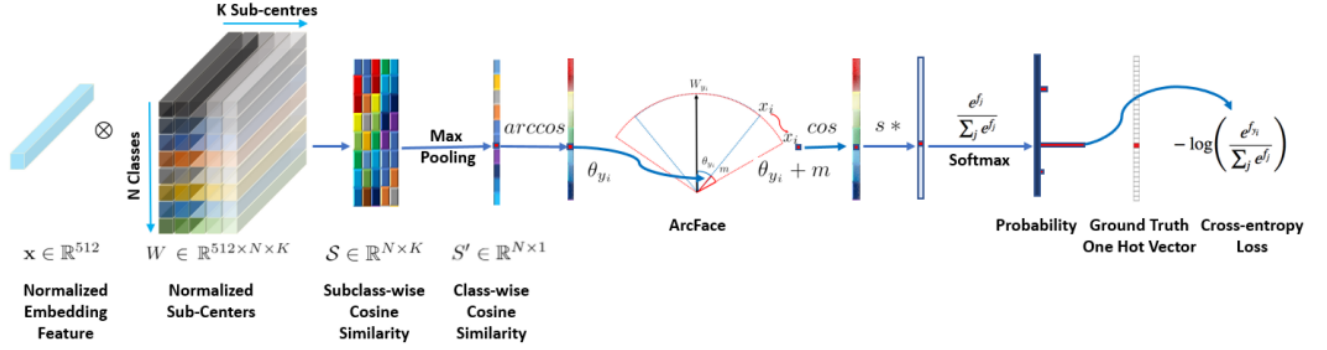


Fig. 2. Arcface

4) **ResNet**: Our model utilizes ResNet (Residual Network) as the core architecture for feature extraction. ResNet's deep structure, with residual blocks, addresses challenges in training very deep neural networks by mitigating issues like vanishing gradients. This allows our model to effectively capture intricate facial features, enhancing performance in face recognition tasks. Notably, we employ a pretrained ResNet model, trained on diverse datasets, ensuring a strong foundation for feature extraction in our application.

B. Implementation

1) **Data loading and pre-processing**: The data set consisting of bollywood actors(2000-2010), cricketers and IITM CS faculty with at least ten images for each person except the case of faculty members. Faces from the images are extracted and saved into subfolders for each class in appropriate dimensions. The images are loaded into memory with their corresponding class labels. Bar charts are plotted for better visualization.

2) **Database generation**: The database consists of face embeddings represented in a 512 dimensional vector form. To generate database, a pre-trained **ResNet** model with **ArcFace** loss is used which was trained on thousands of images. The model takes a 112x112 dimensional image and generates a vector which essentially represents facial features in numerical form. Average embedding vectors are generated for each class and stored in the database in .npy format.

3) **Face recognition**: In the context of our face recognition system, the implementation involves the utilization of a Siamese Network. This neural network is designed to process pairs of facial images and determine their similarity. The process begins

by encoding each facial image into a 512-dimensional vector, creating embeddings within a vector space. These embeddings serve as unique representations for individuals in our database which will be loaded from saved .npy file

When a new facial image is introduced, the Siamese Network generates an embedding for it. Subsequently, the algorithm compares this new embedding with the existing embeddings in our database to ascertain similarity. Successful matches indicate that the introduced facial image corresponds to an individual present in our database, facilitating accurate face recognition. The contrastive loss equation for a Siamese network is given by:

$$L_{\text{contrastive}}(Y, D) = \frac{1}{2N} \sum_{n=1}^N [Y \cdot D^2 + (1 - Y) \cdot \max(\text{margin} - D, 0)^2]$$

The terms represent:

- Y is a label indicating if pairs are similar ($Y = 1$) or dissimilar ($Y = 0$).
- D is the distance between the representations of two things in the network.
- margin is a minimum distance dissimilar pairs should have.
- For similar pairs ($Y = 1$), the loss encourages the network to make their representations close together.
- For dissimilar pairs ($Y = 0$), the loss encourages the network to ensure their representations are at least margin units apart.
- The $\sum_{n=1}^N$ adds up the losses for all pairs in the training data.
- The $\frac{1}{2N}$ normalizes the loss, giving equal importance to each pair.

Its ability to efficiently generate embeddings contributes to the robustness and accuracy of our face recognition system. To measure the similarity between pairs of embeddings, we use the distance norm between the embeddings.

C. Computational Environment and Inference Time

The model's training and testing were executed within the Jupyter Notebook environment, utilizing the system's CPU resources. The CPU specifications for the employed hardware are as follows: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz.

Inference Time:

- Embedding Database Generation of 250 Classes: 16 minutes, equivalent to 3.84 seconds per class.
- Prediction: 0.469 seconds per image.

III. OUTPUT

A three column output with the image, predicted name and confidence score is displayed when a new image is given to the model.




Face	Predicted Name	Confidence Score
	Profsukhendu_Das	38.68
	Asst_Prof_Harish_Guruprasad	50.3
	Prof_D_Janakiram	28.98

Fig. 3. Samples recognized correctly

	Alex_Carey	4.59
---	------------	------

Fig. 4. Sample recognized incorrectly

A. Observation

- accuracy achieved on training data 95%
- accuracy achieved while testing 85%
- A model that wasn't pretrained had to go over many more iterations to converge as compared to the pretrained model.
- Generating and storing embeddings is faster and less resource consuming on comparison to a neural network like CNN. It also achieves higher accuracies with smaller datasets.
- The current model performed better than Ghostnet, Facenet which are other famous models in the usecase of face recognition.

IV. CONCLUSION

In summary, the Face Recognition project successfully integrates a diverse dataset and powerful algorithms, including the Siamese Network, ArcFace, and ResNet, to achieve precise and efficient identification of individuals from web images. The implementation of a sophisticated web scraper ensures the inclusion of high-quality facial images, emphasizing the project's real-world applicability. ArcFace's precision and resilience, combined with ResNet's foundational architecture, create a robust framework capable of handling varying pose and illumination conditions.

Observations indicate commendable accuracy on both training and testing data, efficient embedding generation and storage, and superior performance compared to alternative models. The user-friendly three-column output, featuring the image, predicted name, and confidence score, depicts the practical significance of this project. In conclusion, our Face Recognition project serves as a robust solution, reflecting the practical significance of facial recognition technologies. It demonstrates our understanding and application of key concepts in implementing effective systems for identifying individuals from web images.

REFERENCES

- [1] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition,"[link](#)
- [2] R. Chatterjee, S. Roy, and S. Roy, *A Siamese Neural Network-Based Face Recognition from Masked Faces*, in *Advanced Network Technologies and Intelligent Computing, ANTIC 2021*, I. Woungang, S.K. Dhurandher, K.K. Pattanaik, A. Verma, and P. Verma (eds.), *Communications in Computer and Information Science*, vol. 1534, Springer, Cham, 2022, pp. 1-6; [link](#)
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," *Microsoft Research*, year=2016.[link](#)
- [4] Cricket Players Faces [A Kaggle Dataset](#)
- [5] Indian Cricketer's Images [A Kaggle Dataset](#)
- [6] Indian Actor Images Dataset [A Kaggle Dataset](#)
- [7] Celebs Face Recognition — Facenet [A Kaggle Notebook](#)
- [8] Custom dataset of IITM CS Faculty [link to gdrive](#)