Deep Learning Based Face Recognition Using Clustering in Video for Social Event Media Organization and Sharing

Diana Moses
Associate Professor, Methodist College of Engineering and Technology, Hyderabad, India

Abstract
In the age of digital socializing, promptly sharing personal photos from events has become a priority, creating a demand for quick access to personalized images. Efficiently managing event photos, accurately recognizing attendees, and swiftly delivering personalized images to clients present significant challenges for photographers. Our project addresses these issues by employing advanced machine learning techniques for automated image labeling and recognition. The software we developed facilitates seamless photo management and delivery by leveraging face recognition and clustering methodologies. This enables automatic identification and organization of images, ensuring personalized photos are promptly delivered to clients via email. Our approach aims to streamline photographers' workflows, enhancing their ability to provide fast and customized services to their clients.

Introduction

In recent years, the field of face recognition has seen significant advancements through the integration of deep learning techniques. Wang and Shen (2023) introduced a method for micro-expression recognition based on apex frames using deep learning, showcasing the potential of deep learning in capturing subtle facial cues. This highlights the importance of leveraging deep learning for intricate tasks like facial expression recognition. Additionally, Uzun, Cevikalp, and Saribas (2022) proposed deep discriminative feature models (DDFMs) for set-based face recognition and distance metric learning, emphasizing the role of deep learning in enhancing feature representation for accurate face recognition.

Diana (2020) presents a review of various supervised and unsupervised learning models on images indicates the advantages of different clustering approaches. Trivikram, Diana (2017) presented a hybrid model for recognition of individuals based on both face images and voice in the video snippets. Moreover, the work by Tosidis, Passalis, and Tefas (2022) on active vision control policies for face recognition using deep reinforcement learning demonstrates the application of reinforcement learning techniques in optimizing face recognition systems. This indicates the versatility of deep learning approaches, extending beyond traditional methods to enhance face recognition performance through dynamic control policies. Furthermore, Oinar, Le, and Woo (2023) introduced the Kappaface model, which incorporates an adaptive additive angular margin loss for deep face recognition, showcasing the innovation in loss functions to improve face recognition accuracy.

In the realm of facial expression recognition, Kolahdouzi, Sepas-Moghaddam, and Etemad (2022) presented FaceTopoNet, a model that utilizes face topology learning for robust facial expression recognition. This work highlights the significance of leveraging facial structure information for accurate expression recognition, showcasing the potential of deep learning in capturing nuanced facial features. Additionally, Chi et al. (2023) introduced L-GhostNet, a model designed to extract high-quality features, emphasizing the importance of feature extraction in enhancing face recognition systems.

Furthermore, the study by Liu et al. (2023) on uncertain facial expression recognition via multi-task assisted correction underscores the challenges in handling uncertainty in facial expressions and the potential of multi-task learning in improving recognition accuracy. This work sheds light on the complexities involved in facial expression analysis and the need for robust deep learning models to address uncertainties in expression recognition tasks.

In conclusion, the integration of deep learning techniques in face recognition systems has shown remarkable progress in enhancing accuracy, robustness, and performance. From micro-expression recognition to facial expression analysis and feature extraction, the studies reviewed demonstrate the diverse applications of deep learning in advancing face recognition technologies. These advancements pave the way for more sophisticated and efficient face recognition systems, with implications for various domains such as security, human-computer interaction, and social event media organization and sharing.

Materials Used

The proposed models were tested on 3 different datasets viz LFW, MS-Celeb-1M, IMDb-Face Dataset. The LFW (Labeled Faces in the Wild) dataset is a well-known benchmark for evaluating the performance of face recognition algorithms. It contains a collection of face images designed to reflect the challenges of face recognition in real-world conditions, such as varying lighting, pose, and occlusions.The LFW dataset contains 13,233 labeled images of faces.There are 5,749 unique individuals represented in the dataset.The images were collected from the internet, ensuring a wide variety of conditions, including different poses, lighting conditions, and backgrounds.Each image is labeled with the name of the person. For some individuals, there are multiple images, while for others, there may be only one image. The number of images per individual varies, with some individuals having over a hundred images and others having just one. A sample set from the dataset is shown in figure 1.

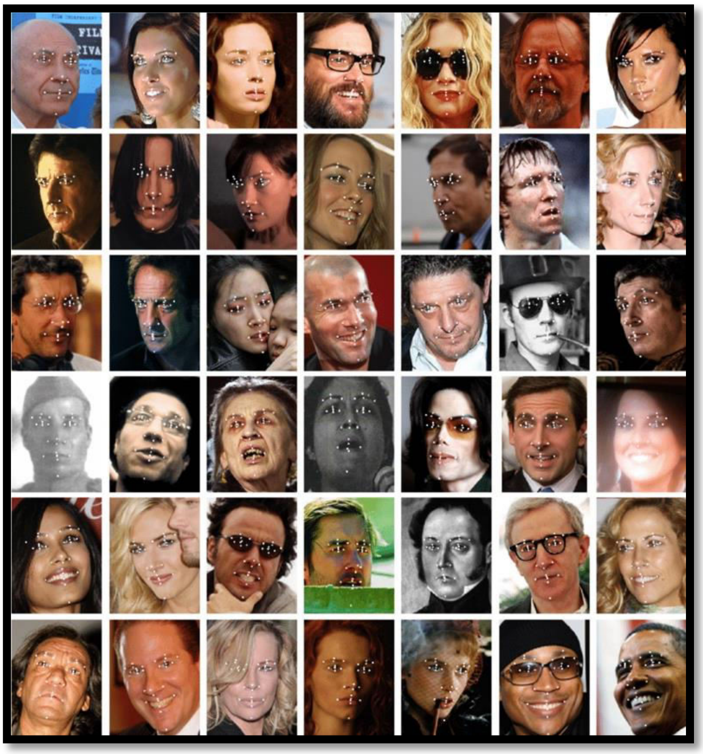


Fig 1. Data samples from LFW Dataset

The MS-Celeb-1M dataset, also known as the Microsoft Celebrity One Million dataset, is a large-scale dataset created for training and evaluating face recognition systems. It contains a vast number of images and identities, making it one of the largest publicly available datasets for face recognition research.The MS-Celeb-1M dataset originally contained around 10 million images. It includes approximately 100,000 unique identities (celebrities). The images exhibit a wide range of variations in terms of pose, lighting, expression, occlusions, and image quality.Each image is labeled with the identity of the person. These identities are primarily celebrities, including actors, athletes, politicians, and other public figures.Each image also comes with metadata such as the URL from where the image was sourced, which can sometimes be used to infer additional information like the context of the image. A sample set from the dataset is shown in figure 2.

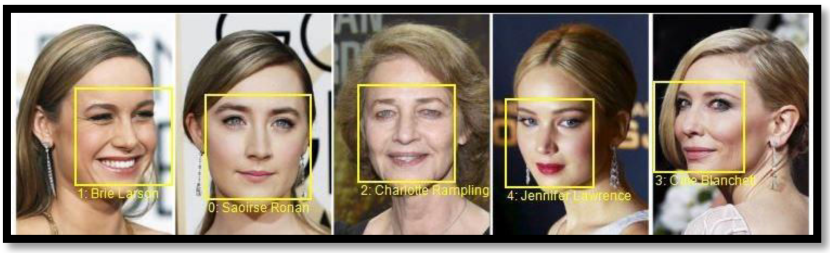


Fig 2. Data samples from MS-Celeb-1M Dataset

The IMDb-Face dataset is a large-scale face dataset designed for training and evaluating face recognition algorithms. It contains images of celebrities, which are collected from the IMDb (Internet Movie Database) website. The dataset is notable for its size and diversity, making it a valuable resource for researchers working on face recognition. The IMDb-Face Datasetcontains around 1.7 million images.There are approximately 59,000 unique identities (celebrities) represented in the dataset.The images include a wide range of variations in pose, lighting, expression, age, and background, reflecting real-world conditions.Each image is labeled with the identity of the person. These labels correspond to the names of celebrities from IMDb.The dataset includes metadata such as the IMDb ID, name, and additional contextual information about the images, which can be used for various analysis and preprocessingtasks.A sample set from the dataset is shown in figure 3.

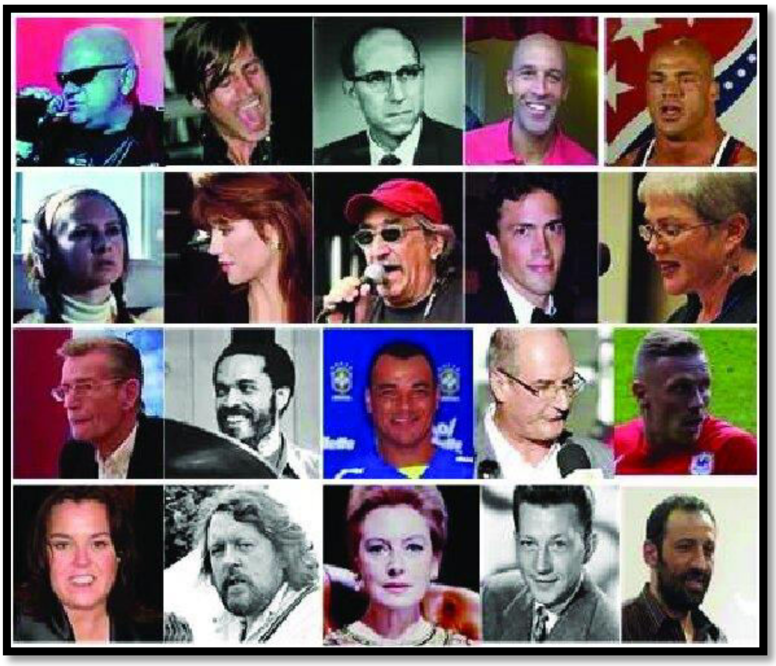


Fig 3. Data samples from IMDb-Face Dataset

Methodolody

Theobjective of the proposed work is to develop a system that automates the recognition of individuals in event photos and organizes them into personalized photo albums for attendees.This methodology outlines the steps involved in developing a deep learning-based face recognition system for organizing and sharing event photos, specifically targeting weddings and similar social gatherings. Each step aims to leverage advanced machine learning techniques to automate and enhance the photo management process. Towards training the face recognition models diverse dataset of photos from social events, particularly focusing on weddings were collected from 3 different datasets explained above. These datasets are annotated with labels for individuals attending the events. The dataset is pre-processed to enhance image quality and ensure consistency.

Face Recognition with Deep Learning: The use of deep learning in face recognition has seen significant advancements. Techniques involving Convolutional Neural Networks (CNNs) and specifically models like Tensorflow have shown high accuracy in identifying and verifying faces. Our project builds on these methods to create robust facial embeddings that can distinguish between different individuals with high precision.
Deep learning-based face detection algorithms (e.g., MTCNN, RetinaFace) are used to detect faces in each photo. The detected faces are aligned to a standardized pose to mitigate variations in lighting, pose, and facial expression. These Deep learning models extract deep facial features using pretrained convolutional neural networks (CNNs) such as VGGFaceandFaceNet, leading torepresentationofeach face in the dataset as a high-dimensional feature vector.

Automated Image Clustering: Clustering algorithms, such as k-means and hierarchical clustering, have been employed in various domains to group similar images together. In the context of face clustering, these algorithms help in organizing event photos by recognizing and grouping faces that belong to the same individual. This approach significantly reduces the manual effort required in sorting photos.Clustering algorithms viz DBSCAN, K-means group similar face embeddings into clusters, representing individual identities. The clusters are refined to handle variations in appearance using spectral clustering and hierarchical clustering. Develop a recognition model that matches detected faces to clustered identities and associate recognized faces with event attendees based on clustering results.

Event Photo Delivery Systems: Previous systems have explored automated photo delivery but often lack the integration of advanced machine learning techniques. Personalized photo albums are created as folders for each attendee containing photos where they appear.Theprocess of creating and packaging personalized photo albums is automated and embedded into the mailing system to send the zipped folders to attendees via email.This seamless integration enhances user experience by providing a fully automated solution.

The efficiency of the face recognition and clustering algorithms is evaluated using the Silhouette score measure. The developed system is deployed as “Pixel Pigeon” in real-world wedding events.

Dataset Utilization: Leveraging large and diverse datasets, such as LFW and VGGFace, our proposed modelensures that the face recognition model is trained on a wide variety of faces, improving its accuracy and reliability. The use of pre-trained models, fine-tuned on event-specific data, allows us to achieve high performance even in real-world scenarios.

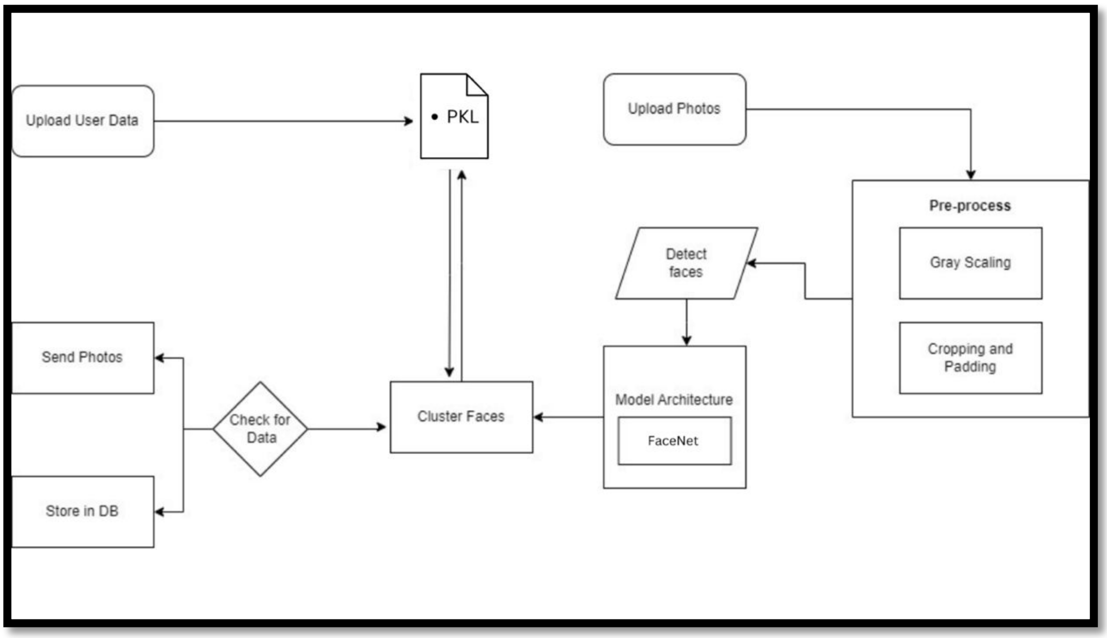


Fig 4. Architecture of the proposed Pixel Pigeon Model

Model Architecture

In this system, we have developed a face recognition and clustering model using FaceNet, which leverages Convolutional Neural Networks (CNNs). During the development phase, we utilized datasets such as LFW, VGGFace, and ImageNet, preprocessing them to create a new processed dataset that included only the face-cropped images.The proposed Pixel Pigeon architecture is shown in Figure 4

The dataset is initially imported using Image Import- Using glob, in the directory into a Python list then preprocessedusing face detection function detect_faces() method from face_recognition library to detect landmarks of face.Frame Cropping is then performed where each detected face is cropped from the image to focus on face locations. Towards maintaining uniform dimensions, all face-cropped images are resized to a resolution of 112 x 112 pixels. These preprocessed images are stored in a new directory, preserving the original format (e.g., JPEG or PNG). The dataset is created by organizing into labelled folders, which is used for further face recognition and clustering.This ensures that all images are uniformly processed, making the subsequent face clustering and recognition steps more efficient and accurate.

To efficiently deliver personalized event photos, it is essential to understand the underlying mechanisms of face recognition and clustering. Our approach involves training the FaceNet model on datasets with a diverse range of identities. FaceNet takes a source image as input, processes it to detect and align the face region, and then passes it through a CNN model to extract relevant facial features. This process generates a fixed-length vector (e.g., 128-D for FaceNet) representing the facial embedding or feature vector. These embeddings are then used to cluster faces, enabling the automated sorting and distribution of event photos.

The trained model can recognize and group faces with high accuracy, ensuring that each event attendee receives their personalized photos via email in a convenient zip file. The objective of this system is to streamline the photo delivery process, eliminating the need for manual sorting and enhancing the user experience by providing timely and accurate photo delivery.Tools for Face Recognition and Clustering used are FaceNet, Dlib, OpenCV, Scikit-learn, Python face_recognition library.

Our model is based on the FaceNet architecture, utilizing Convolutional Neural Networks (CNN) for feature extraction and clustering. FaceNet is known for generating 128-dimensional embeddings for each face, allowing for efficient recognition and clustering of faces within videos.

We used the pre-trained FaceNet model for feature extraction. FaceNet is a sophisticated CNN model optimized for face recognition tasks. It maps facial images to a compact Euclidean space where distances correspond to a measure of face similarity. The 128-dimensional embeddings produced by FaceNet represent these facial features in a highly discriminative manner.For our purposes, we fine-tune FaceNet by adding additional layers and selecting an appropriate learning rate to ensure the model effectively converges during training. The embeddings generated by FaceNet serve as input for further processing and clustering.

The 128-dimensional embeddings from FaceNet are used as input for clustering algorithms to group similar faces together. This allows the system to recognize and categorize different faces appearing in the video. We employ algorithms such as DBSCAN or K-Means for the clustering process.The model incorporates the Leaky ReLU activation function and additional linear layers to refine the embeddings. A batch size of 32 is utilized during training to balance computational efficiency and model performance. A SoftMax layer is applied to obtain the model's confidence levels during predictions.

Hyper-parameter tuning is crucial for achieving optimal model performance. After multiple iterations, we identified the best hyper-parameters for our dataset. We used the Adam optimizer with adaptive learning rates, starting at 1e-4 (0.0001) to achieve a better global minimum for gradient descent. The weight decay parameter is set to 1e-4.Given that our task involves classification, we used cross-entropy loss to measure the model's performance. Batch training is employed to make efficient use of computational resources, with a batch size of 32 proving to be ideal for our environment.

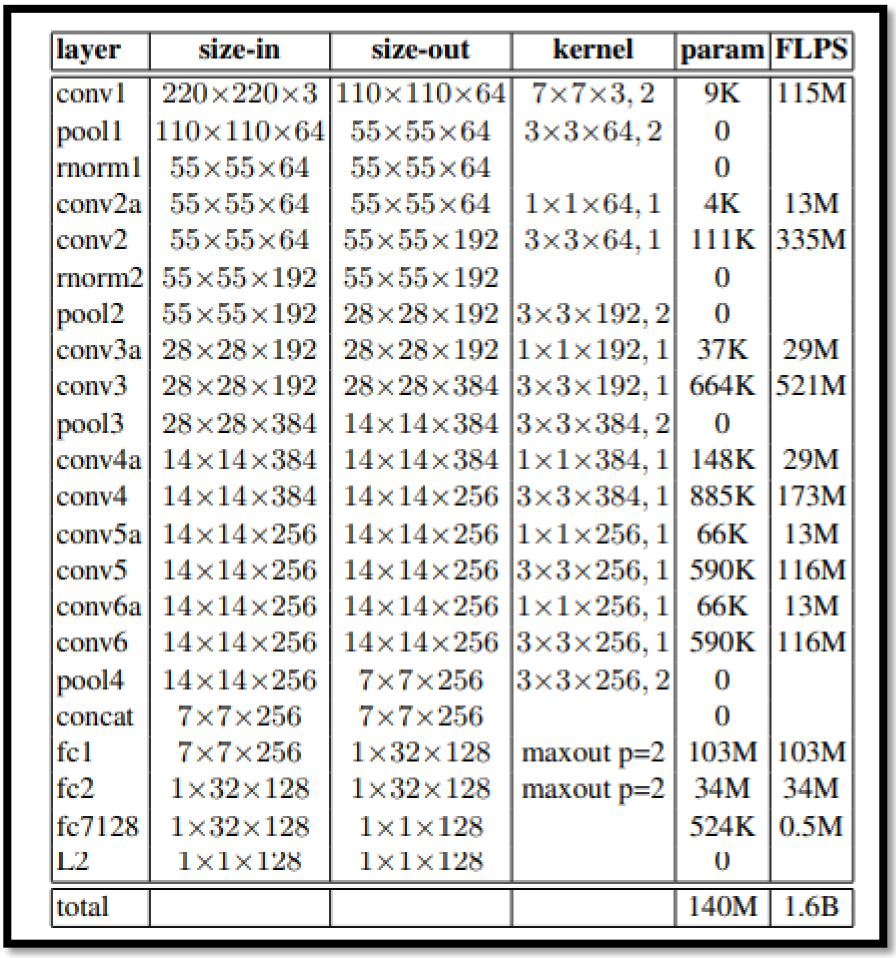
The User Interface (UI) for our application is developed using the Flask framework. The main page, index.html, includes a tab for browsing and uploading videos. Once a video is uploaded, it is processed by the model, which predicts whether the video is real or fake and provides a confidence score. The results are displayed in predict.html, where the video is played with detected faces highlighted and annotated with the prediction and confidence score.

Evaluation metrics can help you assess your model’s performance, monitor your machine learning system in production, and ensure your model meets your business needs. Our goal is to create and select a model that gives high accuracy on out-of-sample data. It's crucial to use multiple evaluation metrics to evaluate your model because a model may perform well using one measurement from one evaluation metric while performing poorly using another measurement from another evaluation metric.

Model Details

The model consists of following layers:

FaceNet CNN: The pre-trained FaceNet model is utilized for extracting facial features. FaceNet generates 128-dimensional embeddings for each face, providing a compact and highly discriminative representation of facial features. This model uses a deep CNN architecture optimized for face recognition tasks. The details of the layers are detailed in figure 5.



layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Figure 5.FaceNet Architecture

FaceNet uses multiple layers of convolutions, activations (like ReLU), and pooling. The network processes an input image through these layers to extract features. The final layer before the embedding is a fully connected layer, which maps the features to a high-dimensional space. The output of this layer, after applying the identity function, is the face embedding. During training, weights are adjusted to minimize the triplet loss, ensuring that embeddings of the same person are closer together than those of different people. The working of FaceNet is presented in Figure 6.

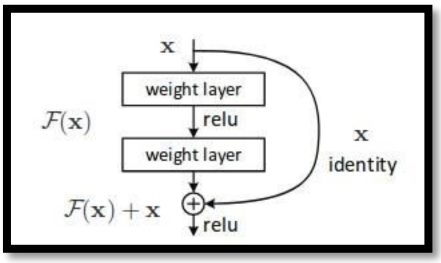

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
x
weight layer
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 6.FaceNet Working

The Sequentialis a container of Modules that can be stacked together and run at the same time. The sequential layer is used to store the feature vector returned by the facenet model in an ordered way. So that it can be processed for clustering and recognition. The ReLU (Rectified Linear Unit) is an activation function that has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. The operation of ReLU is closer to the way our biological neurons work. The ReLU function is shown in figure 7. ReLU is non-linear and has the advantage of not having any backpropagation errors, unlike the sigmoid function, also for larger Neural Networks, the speed of building models based on ReLU is very fast.

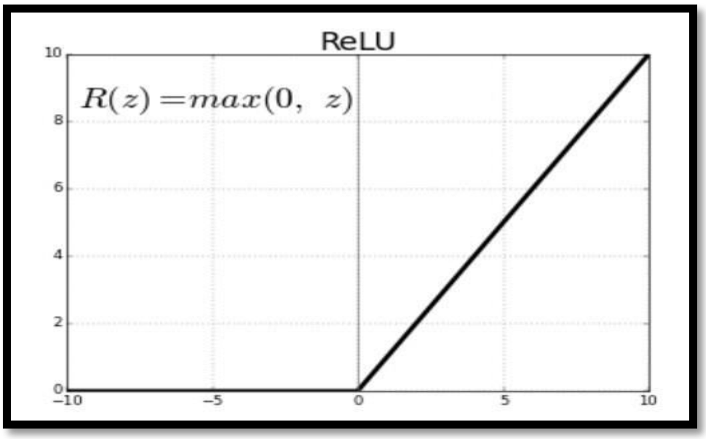


Figure 7.Relu Activation function

Following is a Dropout Layerwith the value of 0.4 which is used to avoid overfitting in the model and it can help a model generalize by randomly setting the output for a given neuron to 0. The connectivity of the dropout layer is illustrated in fig 8. In setting the output to 0, the cost function becomes more sensitive to neighbouring neurons changing the way the weights will be updated during the process of backpropagation.

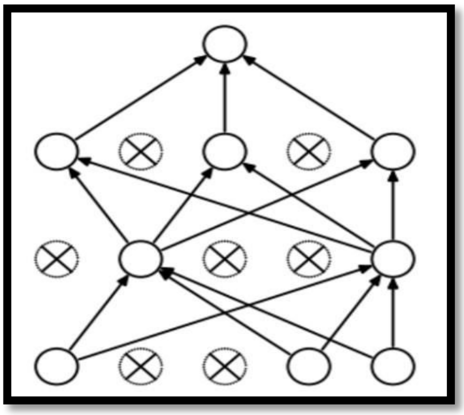


Figure 8: Dropout layer

There is an Adaptive Average Pooling Layer which is used to reduce variance, reduce computation complexity and extract low level features from the neighbourhood.2-dimensional Adaptive Average Pooling Layer is used in the model.

Model Training and Testing

The dataset is split into training and testing datasets with a ratio of 70% for training and 30% for testing. The split is balanced, ensuring that both the training and testing sets contain an equal proportion of images from different classes.Data Loaderis used to load the images and their corresponding labels with an appropriate batch size for training.Training is performed for 20 epochs with a learning rate of 1e-4 (0.0001) and a weight decay of 1e-4 (0.0001) using the Adam optimizer.Adam Optimizer is employed to adapt the learning rate during training, providing efficient and effective convergence.Cross-Entropy Loss is used as the loss function since the task is a classification problem.The SoftMax layer is used as the final layer to obtain the probability distribution over the possible classes. The output of the SoftMax layer can be interpreted as a probability, with the sum of probabilities equaling 1. In this project, the SoftMax layer provides the confidence level for each prediction, ensuring accurate classification and recognition of faces. The model trained is valdated using the silhouette score.

Silhouette Score

The Silhouette score is a metric used to evaluate the quality of a clustering algorithm. It measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The Silhouette score combines two factors for each data point:

1. **Cohesion (a)**: The average distance between a data point and all other points in the same cluster.
2. **Separation (b)**: The average distance between a data point and all other points in the nearest cluster (i.e., the cluster that the point is not a part of but is closest to).

The Silhouette score for a single data point **i**is calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

The value of s(i) ranges from -1 to 1.An s(i) of**1** indicates that the data point is well-matched to its own cluster and poorly matched to neighboring clusters. An s(i) of**0** indicates that the data point is on or very close to the decision boundary between two neighboring clusters. An s(i) of-1 indicates that the data point might have been assigned to the wrong cluster.

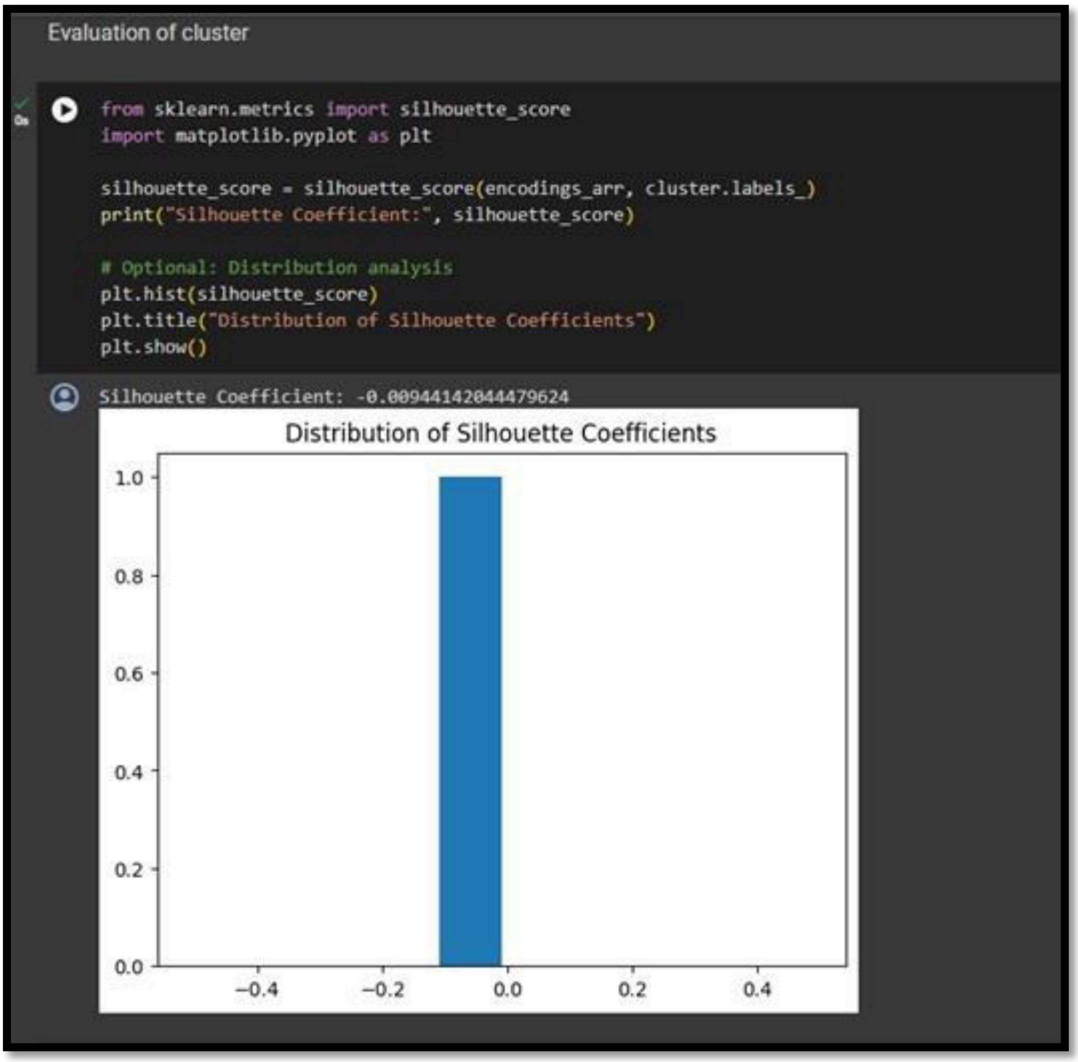
The overall Silhouette score for a clustering solution is the average Silhouette score of all the data points. It provides an indication of the overall quality of the clustering. A higher average Silhouette score suggests better-defined clusters.

Steps to Compute Silhouette Score

1. **Compute cohesion**a(i): For each point, calculate the average distance to all other points in the same cluster.
2. **Compute separation**b(i): For each point, calculate the average distance to all points in the nearest cluster.
3. **Calculate Silhouette scores**s(i): For each point, use the formula as given above.
4. **Average the scores**: Compute the mean of the Silhouette scores for all points to get the overall score.

$$\text{Silhouette Score} = \frac{\sum_{i=1}^{n} s(i)}{n}$$

The Silhouette score of the proposed face clusreing models is presented in Figure 9.



Evaluation of cluster

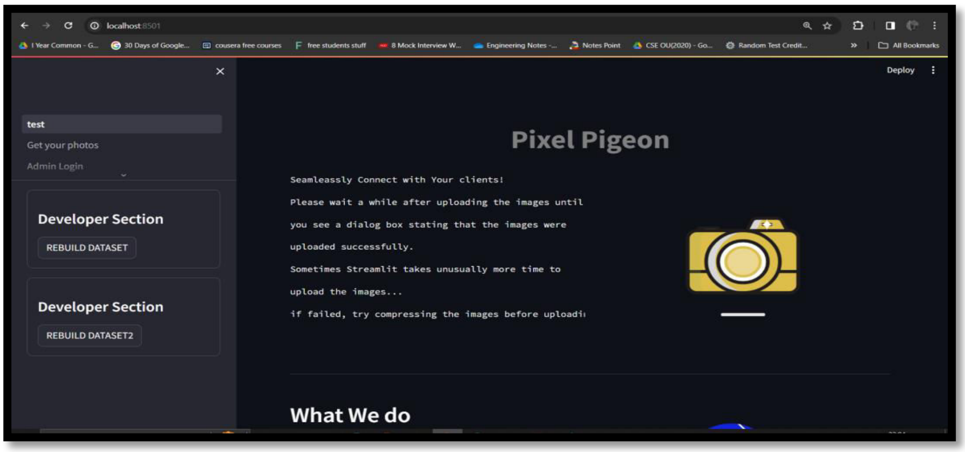
```
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

silhouette_score = silhouette_score(encodings_arr, cluster.labels_)
print("Silhouette Coefficient:", silhouette_score)

# Optional: Distribution analysis
plt.hist(silhouette_score)
plt.title("Distribution of Silhouette Coefficients")
plt.show()
```

```
Silhouette Coefficient: -0.009441420444479624
```

Figure 9. Silhouette score for the proposed model

Developing a user-friendly interface for event organizers and attendees to interact with the system, ensuring ease of use and seamless photo delivery id the objective of the proposed system. Our solution has the potential to be scaled up and integrated with larger platforms such as event management software or social media networks, making it a versatile tool for various applications. By automating the tedious task of photo sorting and delivery, "Pixel Pigeon" aims to enhance the event experience for attendees and streamline the workflow for event organizers.

The user interface desing and delivery of photographs identified by the clustering model to respective attendees of the events are shown in figure 10, 11 and 12.



Pixel Pigeon

Seamleassly Connect with Your clients!

Please wait a while after uploading the images until

you see a dialog box stating that the images were

uploaded successfully.

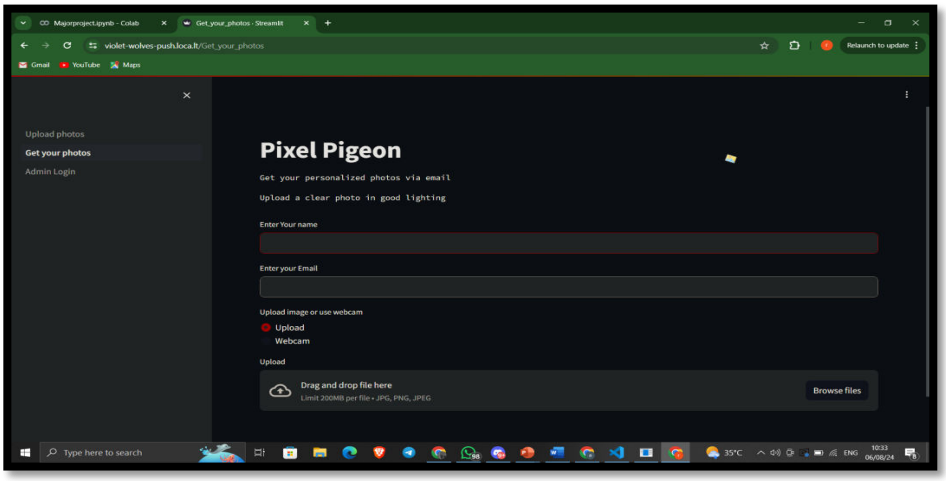
Sometimes Streamlit takes unusually more time to

upload the images...

if failed, try compressing the images before uploadi

What We do

Figure 10. Pixel Pigeon User interface design



Pixel Pigeon

Get your personalized photos via email

Upload a clear photo in good lighting

Enter Your name

Enter your Email

Upload image or use webcam

● Upload

○ Webcam

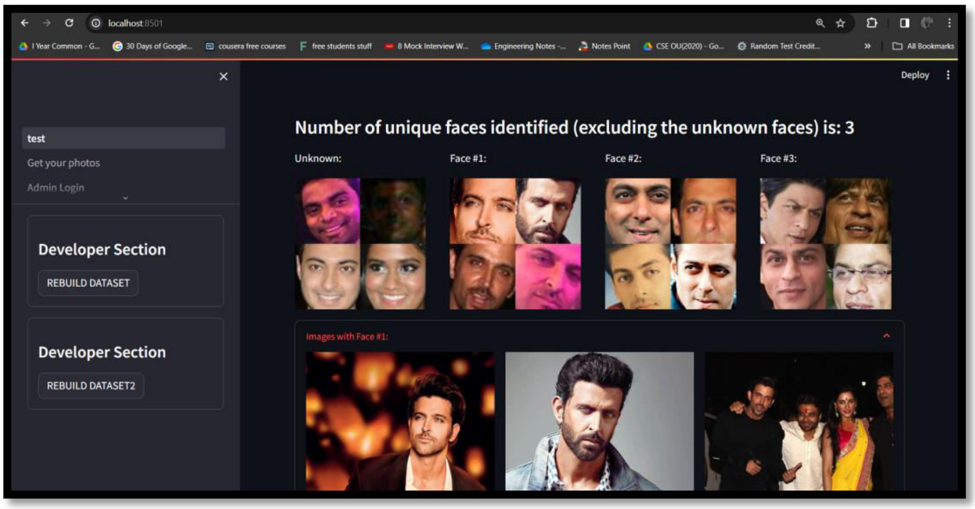
Upload

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

Browse files

Figure 11. Pixel Pigeon User input module



Number of unique faces identified (excluding the unknown faces) is: 3

Unknown:

Face #1:

Face #2:

Face #3:

Images with Face #1:

Figure 12. Pixel Pigeon Clustered faces

Conclusion

We developed a neural network-based system for automated photo delivery using face clustering and recognition. Our method efficiently identifies and clusters faces from event photos, leveraging FaceNet and Convolutional Neural Networks (CNNs). The system preprocesses images, detects and aligns faces, and generates facial embeddings for clustering, ensuring accurate and timely photo distribution to attendees via email. By using pre-trained models, our approach maintains high accuracy in face recognition and performs reliably throughout the event photo delivery process. Enhancements can further improve the system, especially with evolving technologies and new opportunities by expanding the application into a web-based platform for broader accessibility and convenience, including group photo detection and clustering for comprehensive event photo management and integrating with social media platforms for direct photo sharing, increasing utility and user engagement.

References

1. Wang, Xianhua, and XunbingShen. "Micro-expression Recognition Based on Apex Frame Using Deep Learning." 2023 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). IEEE, 2023.
2. Trivikram, Cheluri, Diana. "Evaluation Of Hybrid Face And Voice Recognition Systems For Biometric Identification In Areas Requiring High SecuritY." I-Manager's Journal of Pattern Recognition 4.3 (2017).
3. Moses, Diana. "Review on Challenges of Machine Learning in Future of Healthcare." Solid State Technology (2020): 7262-7282.
4. Tosidis, Pavlos, NikolaosPassalis, and AnastasiosTefas. "Active vision control policies for face recognition using deep reinforcement learning." 2022 30th European Signal Processing Conference (EUSIPCO). IEEE, 2022.
5. Uzun, Bedirhan, HakanCevikalp, and HasanSaribas. "Deep discriminative feature models (DDFMs) for set based face recognition and distance metric learning." IEEE Transactions on Pattern Analysis and Machine Intelligence 45.5 (2022): 5594-5608.
6. Oinar, Chingis, Binh M. Le, and Simon S. Woo. "Kappaface: adaptive additive angular margin loss for deep face recognition." IEEE Access (2023).
7. Li, Wei, et al. "Face Recognition Model Optimization Research Based on Embedded Platform." IEEE Access 11 (2023): 58634-58643.
8. Chi, Jing, et al. "L-GhostNet: Extract better quality features." IEEE Access 11 (2023): 2361-2374.

9. Liu, Yang, et al. "Uncertain facial expression recognition via multi-task assisted correction." IEEE Transactions on Multimedia (2023).
10. Kolahdouzi, Mojtaba, AlirezaSepas-Moghaddam, and Ali Etemad. "FaceTopoNet: Facial expression recognition using face topology learning." IEEE Transactions on Artificial Intelligence 4.6 (2022): 1526-1539.
11. Jabberi, Marwa, et al. "Face shapenets for 3d face recognition." IEEE Access 11 (2023): 46240-46256.