Facial and Emotion Recognition: A Unified Approach

Abstract

This project proposal outlines the creation of machine learning models for precise emotion classification and facial recognition. Given the larger team size, the project is strategically divided into two segments: facial recognition and emotion recognition. The team is evenly distributed, with one half dedicated to developing facial recognition models and the other half focusing on emotional recognition. The overarching object is to create two distinct models to accurately predict both the emotion and identity of an individual.

Introduction

1.1 Biography

Alan Leon (leonalan) – 3rd year Computational Data Science student.

Spencer Dork (dorkspen) – 3rd year Computational Data Science student.

Ryan Hanks (hanksrya) – 3rd year Computational Data Science student.

Aryan Sharma (sharm152) – 3rd year Computational Data Science student.

Mateja Milicevic (milicev2) – 3rd year Computational Data Science student.

Ethan DeMott (demotet) – 3rd year Computational Data Science student.

In recent years, artificial intelligence and machine learning has paved the way for groundbreaking advancements in computer learning. Facial recognition has emerged as a cornerstone, finding broad applications in security, marketing, and personal device authentication, highlighting its importance in modern day applications. Our project aims to elevate this technology by integrating facial recognition with emotion detection, enhancing the model. Leveraging diverse algorithms, we anticipate a synergists effect that will improve our system's ability to accurately identify and interpret facial expressions across a diverse range of contexts and individuals. Our methodology will let us combine the predictive capabilities of various models, reducing the likelihood of errors and increasing our recognition system's reliability. Through careful experimentation and optimization, we are confident that this strategy will yield a more robust and effective solution tailored for real-world scenarios. Lastly, the group will work to ensemble the models together either with fair influence on prediction or weights depending on each model's performance. Additionally, the group has outlined the planned division of labor below.

Division of Labor:

- <u>Facial Recognition Model:</u> Alan Leon, Aryan Sharma, Ethan DeMott
  - Type of model: VGG, Flat Augmentation with some Combination Augmentations, and Histogram equalization
- <u>Facial Emotion Detection:</u> Spencer Dork, Ryan Hanks, Mateja Milicevic
  - Type of model: CNN, VGG, OpenFace

Related Works

Facial recognition software has become an integral component of modern technology, finding applications beyond mobile devices and security measures. In addition to safeguarding personal data, businesses utilize facial recognition in customer service interactions, personalizing user experiences and streamlining processes. Moreover, advancements in this technology have extended its utility to healthcare, where facial recognition aids in patient identification and medical record management, illustrating its versatility in addressing a variety of societal needs. However, facial recognitions have many ethical considerations for all of these uses and this project does not prioritize the in-depth exploration of these implications.

Emotion detection technology plays a crucial role in modern society, offering multifaceted applications beyond its initial scope. Beyond its primary function of enhancing user experiences in digital interfaces, emotion detection technology contributes significantly to various societal domains. Businesses utilize emotion detection in customer service interactions, tailoring responses based on customer sentiment and enhancing overall satisfaction. In healthcare, emotion detection aids in patient care by enabling healthcare providers to gauge emotional states and respond accordingly, thereby improving patient outcomes and overall well-being. Additionally, in educational settings, emotion detection technology can assist educators in understanding student engagement levels and adapting teaching strategies to optimize learning experiences. While ethical considerations surrounding emotion detection persist, its potential to address societal needs remains undeniable.

Diverging from cultural endeavors, this project introduces a distinctive approach by concurrently addressing two facial recognition tasks: identifying the individual and discerning their displayed emotion. Moreover, our unique and distinct methodology distinguishes itself through the incorporation of multiple machine learning techniques to enhance the predictive capabilities. Collaboratively, our team aims to elevate the complexity and overall prediction accuracy, aspiring to mitigate the occurrence of false matches when employing facial detection software. This innovative undertaking seeks to push the boundaries and contribute to the continual improvement of facial recognition technology.

In the exploration of emotion recognition technologies, a notable GitHub project named DeepFace created by Serengil emerges as a comprehensive framework integrating various facial recognition and attribute analysis methodologies. Utilizing a robust set of facial recognition algorithms, including VGG-Face, Google FaceNet, Facebook-DeepFace, OpenFace, DeepID, ArcFace, Dlib, and SFace, DeepFace establishes a solid foundation for identifying individuals. These algorithms are renowned for their precision and efficiency in processing facial features and have “surpassed human-level accuracy”.

Beyond mere facial recognition, DeepFace extends its capabilities to the analysis of facial attributes, utilizing OpenCV, SSD, MTCNN, Dlib, RetinaFace, MediaPipe, Yolo, and YuNet. This multifaceted approach enables the system to detect and analyze various attributes, including age, gender, and emotion, with a notable focus on emotion recognition. The inclusion of race and ethnicity prediction further broadens the project's scope, showcasing its potential to deliver nuanced insights into human facial characteristics.

The integration of these advanced technologies positions DeepFace as a significant contributor to the field of emotion recognition. Its ability to harness the strengths of multiple facial recognition and attribute analysis tools not only enhances its performance but also underscores the project's relevance in ongoing research and applications. As emotion recognition continues to gain importance in various domains, including security, marketing, and human-computer interaction, DeepFace's comprehensive approach offers valuable insights and capabilities to researchers and practitioners alike, pushing the boundaries of what is possible in understanding and interpreting human emotions through technology.

Data Set and Methodology

1.2 The Datasets

Facial Recognition Dataset:
https://drive.google.com/drive/folders/1TR4QkosPsngfLoCVDuN9cn89hQnBU0qE?usp=drive_link

The facial recognition dataset consists of facial photographs of various celebrities. We found this dataset easy to work with as most of the photos are clear, have good lighting and the celebrities are always facing in a general direction of the camera. Another reason is that since these photos are publicly available it was much easier to work with.

Emotion Detection:
https://www.kaggle.com/datasets/ananthu017/emotion-detection-fer

Emotion detection dataset that was acquired from a Machine Learning Forum/Competition Platform contains images of people experiencing 7 different emotions. Emotion labels in this dataset are: angry, disgusted, fearful, happy, neutral, sad and surprised. In total there are 35686 images in this dataset, all of size 48 by 48 pixels. Here is an example of a angry image:

Some of the methodology will be different for each part of the project due to data type limitations. The emotion dataset will be only black and white images so all the color augmentations for facial recognition will not be needed. The differences are noted.

1.3 Prepare the Training Data

In addressing the facial recognition challenge, our approach involves dividing the dataset into distinct training and testing subsets. We adopt a conventional 75/25 split, allocating 75% of the data to the training set for model training and the remaining 25% for evaluation. By using such split, we ensure no overfitting on our training data. Conversely, for the emotion detection task, the dataset arrives pre-divided into predefined training and testing partitions. Therefore, our focus shifts towards developing a robust function tasked with efficiently loading images from their designated directories, streamlining the subsequent model development process.

1.4 Process the Training Data

Normally in computer vision problems, you would resize your images as they hold relatively the same amount information with smaller resolutions. That way we also make the training process faster. For facial recognition, we would scale down all our images to 224x224 pixels. For emotion detection all the images were already scaled to a 48x48 resolution. We believe that this severely reduced our accuracy as we would not go below 224x224. To combat this there were considerations of using resolution enhancing methods such as super resolution but there was a general belief that this would introduce a lot of noise to our data, thus hurting our model instead of helping it. Lastly, to make our Facial Recognition model generalize better we employed augmentation. Augmentation is a method of artificially increasing your training dataset by applying various effects to your images that make them slightly different. For our case we employed 9 augmentations, 5 single and 4 double augmentations. For single augmentations we used grayscale, darken, contrast, saturation, and horizontal flip. For double augmentations we used a combination of horizontal flip and any of the single augmentations. Since most of our images were clean, and the scope of the project was to develop a model that can classify accurately for the datasets we used, we did not employ artificial noise augmentations.

1.5 Model Structure

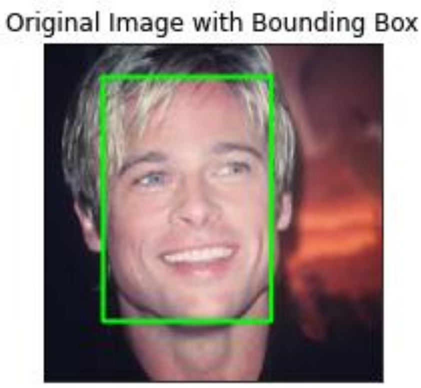
In our study, we used three models in total: two for facial recognition and one for emotion detection. All our models were made and trained using the Python library TensorFlow.

For facial recognition, we trained two VGG16 models with slight variations in training. We modified the VGG16 model to have outputs that are in line with our dataset by adding a Dense output layer of width 17 with a SoftMax activation function. SoftMax activation takes the outputs of the last layer and applies the following function to it:

$$Softmax(z) = \frac{e_i^z}{\sum_{j=1}^{n} e_j^z}$$

By applying the SoftMax function, the outputs of our last layer are always between 0 and 1 as we need probabilities. We would then apply argmax to the output vector, so we know which class has the highest probability for the given image.

For the preprocessing step, before feeding the images into the model, we utilized a face detection algorithm to locate and extract the facial region. This involved creating bounding boxes around detected faces, which serves as the input for our facial recognition models. We employed techniques such as Haar cascades and MTCNN to achieve accurate face detection and extraction. As an example, shown next is the original image of Brad Pitt and then the image with the bounding box.



For emotion recognition, we created a model from scratch. The model consists of three convolutional layers followed by a fully connected Dense network consisting of three layers. Convolutional Neural Networks were first introduced by Yann LeCun and ever since they have been the backbone of computer vision. Convolutional layers perform a convolution on the input data, which is applying a sliding window filter or kernel to produce a feature map. This way Convolutional networks can highlight important features in the image such as edges, textures or shapes. Recently, Transformer based models such as ViT were able to outperform convolutional networks in image classification tasks. They were in consideration for this project, but due to lack of computational resources required for training such model, we decided to refrain from them and stick with Convolutional Neural Networks.

Normally in a CNN, each convolutional layer is followed by pooling and batch normalization. The pooling layers are important as they allow us to reduce the dimensions of our input as we apply convolutions to it. There are two types of pooling, max pooling and average pooling. Max pooling selects the maximum number of elements from the region that the pooling is applied to. This way, the CNN knows how to use its filter so that the most important features will have the highest values. Average pooling on the other hand takes the average. Pooling layers are also very beneficial as they reduce overfitting by providing an abstracted form of our initial input. After pooling we have the batch normalization. With batch normalization, we normalize the data exiting the pooling layers such that they are in the range of –1 to 1 with a mean of 0. Normalization layers are meant to reduce sensitivity to initialization of the weights, have a regularization effect, improve performance, and lastly accelerate the training by allowing for higher learning rates. After the convolutional layers, the output is flattened before entering the dense layers. It is necessary to do that as the fully connected layer only takes one-dimensional data. In the fully connected network, the first two Dense layers are followed by the ReLU activation function and Batch Normalization, where the last Dense layer is followed by SoftMax activation. The ReLU activation is similar to SoftMax and is used widely in Neural Networks as a form of introducing nonlinearity to our model. ReLU works by setting all the negative values in the output to zero while preserving the positive values. The following is the ReLU function:

$$Relu(z) = \max(0, z)$$

1.6 Train the Model

For training the Facial Recognition models, we implemented a rigorous approach using K-Fold Cross-Validation with 10 folds. This method allowed us to partition our dataset into 10 subsets, ensuring that each subset was used as both a training and validation set across different iterations. By utilizing K-Fold Cross Validation, we aimed to prevent our models from overfitting to the training data, thereby enhancing their generalization performance. K-Fold cross-validation achieves that by training a new model on each of the folds and saving the best model to be the initial model for the next epoch. In general, in most data science projects K-Fold Cross-validation has shown superior performance to other methods such as Hold-out Validation, leave-one-out validation, and others. Across all our models, we employed Adam optimization, a popular and effective optimization algorithm known for its robustness and efficiency in training deep neural networks. Similar in the way it works to Stochastic Gradient Descent, we chose to go with Adam for a few reasons. By experimentation, it was shown that the Adam optimizer converges faster and achieves lower loss values, which stands for Adaptive moment estimation, and achieves its great performance by computing adaptive learning rates from estimates of the first and second moments of the gradient. This addition to the learning rate mechanism, makes the Adam optimizer different in that regard to the Stochastic Gradient Descent, as SGD uses a fixed learning rate. We utilized categorical cross entropy as the loss function, a widely used metric for multi-class classification tasks. This choice of loss function helped us effectively measure the disparity between the predicted and actual class labels, guiding the optimization process toward minimizing classification errors. Next, the process for training specifically for facial recognition will me discussed.

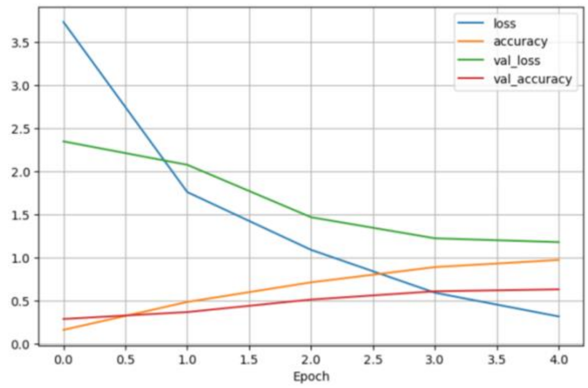
Moreover, during the model compilation the model is trained using LOOCV but also with the augmentation images. However, the augmentation images are never a part of the validation set only for training and these 9 augmentations are applied to every single image for training using an apply augmentations function, so for each training fold the with number validation images being 180 which is the size of the Celebrities Dataset divided 10 for 10 number of training images are ((1800-180)+(9*(1800-180))=16200) where (9*(1800-180)) are augmented images.The list of the performed augmentations for this model include:

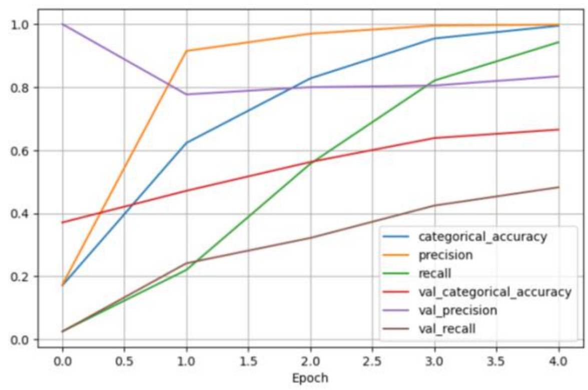
1. Applying Grayscale
2. Applying Darken (Shadow effect)
3. Increasing Contrast
4. Increasing Saturation
5. Horizontal Flip
6. Horizontal Flip and Grayscale
7. Horizontal Flip and Darken
8. Horizontal Flip and Contrast Increase
9. Horizontal Flip and Saturation Increase

The reason we utilized these augmentation techniques was we found that these augmentations to be optimal after trial and error as they did not harshly modify the images to the point where they are unrecognizable to the model, but they are just enough to enhance the prediction accuracy of the model. The reason we only used horizontal flip for combination augmentations was not only for image recognizability reasons, but because we had found it the best way to manipulate the orientation of the faces in the images within our dataset in contrast to methods like shearing which were more of detriment than an enhancement. Below is an image displaying a varying range of augmentations we had tried for previously run models:

In terms of training epochs, our Emotion Detection model underwent training for 10 epochs, allowing it to iteratively learn and adjust its parameters to better capture the nuances of emotional expressions in the dataset. On the other hand, our Facial Recognition models were trained for 5 epochs, striking a balance between computational efficiency and model convergence. These carefully selected numbers of epochs ensured that our models had sufficient training iterations to converge to optimal solutions without risking overfitting or excessive computational burden. As we can see with this number of epochs, we were able to converge our model without overfitting it.

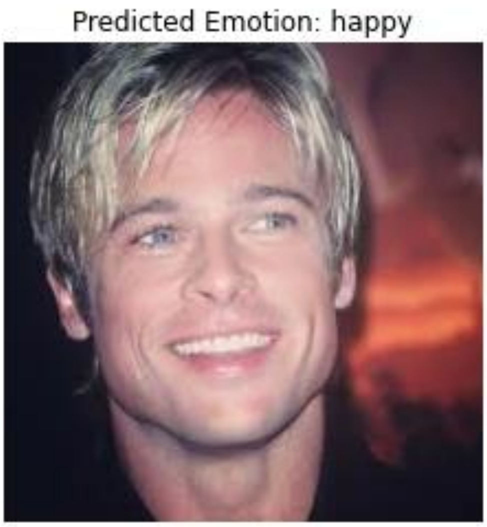
The first plot showcases the loss and accuracy scores of both the training and testing datasets. Our second plot illustrates how various metrics (accuracy, precision, and recall) change as the number of epochs increases.





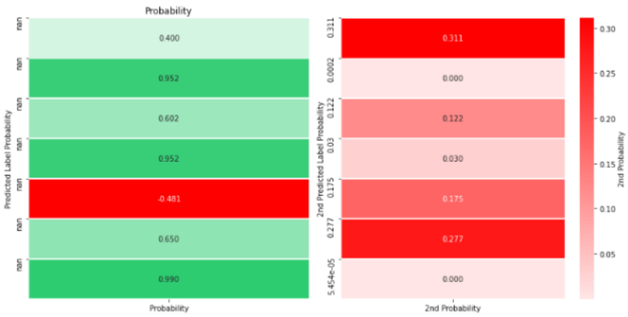
1.7 Test the Model on Validation Set

Following the completion of the training phase, each of our models underwent testing using images sourced from an external dataset, namely the validation set. This validation process is crucial as it allows us to assess the models' performance on unseen data, thereby providing a robust measure of their generalization capabilities. By subjecting our models to this validation procedure, we ensure their reliability and effectiveness in real-world scenarios beyond the confines of the training data. Shown below is the predicted probability and predicted label, that being of Brad Pitt, on the facial recognition model with no ensemble. Also shown below is the predicted emotion of the same picture of Brad Pitt. Both models performed well, correctly classifying the face and the emotion.

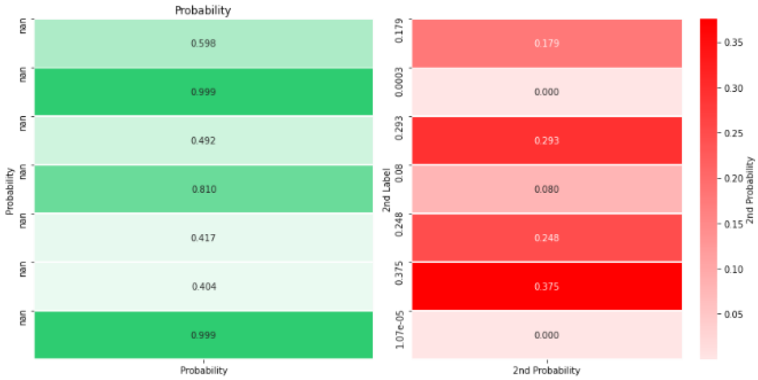


1.8 Utilize Ensemble of Models

In our ensemble approach, we utilized the ensemble technique exclusively for our facial recognition models. When considering the model without ensemble, most predicted labels were accurate, with one exception observed in the image "pitt_bald_test.jpg." In this instance, the model misclassified the image as Denzel Washington with a probability of 0.481, as indicated by the red representation in the first heatmap. Consequently, the model without ensemble achieved an accuracy of 6 out of 7 for the 7 test images. Analyzing the second heatmap, it reveals probabilities associated with the second highest predicted labels, generally hovering around an average of 0.13 probability. This is desirable, as higher probabilities should be allocated to the correct label, while lower probabilities are preferred for alternative labels. Notably, in the misclassified image, the second highest predicted label was Robert Downey Jr., indicating that the correct label was not among the top two predictions, but rather emerged as the third. This observation underscores the necessity of employing ensemble techniques over training a standalone model.



Contrastingly, the ensemble model yielded accurate predictions for all test images, achieving a perfect score of 7 out of 7. Like the previous model, analysis of the second heatmap indicates probabilities associated with the second highest predicted labels. On average, these probabilities slightly increased to 0.167 compared to the model without ensemble, albeit remaining relatively low. However, it's crucial to note that one portion of the ensemble, which excludes histogram equalization as a preprocessing step, misclassified the image "pitt_bald_test.jpg." Conversely, the other portion, incorporating histogram equalization, correctly classified this image but misclassified two others: "portaman_test.jpg" and "jolie_hair_eyes_test.jpg." By averaging the probability results of these two ensembled models, achieving accuracies of 6 out of 7 and 5 out of 7 respectively, we were able to achieve a perfect accuracy of 7 out of 7.



1.9 Results

We assessed the performance of our models by measuring their accuracy on the testing dataset. The results are shown below with for Emotion Detection (ED) and the two Facial Detection (FD) models.

	ED	FR1	FR2
Precision	53.45%	72.94%	83.46%
Recall	52.17%	63.86%	48.22%
Accuracy	51.62%	67.78%	66.44%
F_1 Score	52.09%	68.10%	61.25%

1.10 Conclusion

Moving forward, there are several avenues for enhancing our models and improving their performance. One such option is the utilization of Generative Adversarial Networks (GANs) to augment our dataset. GANs can generate synthetic facial images that closely resemble real ones, thereby expanding our training data and potentially improving the robustness of our models. Additionally, integrating more diverse models into our ensemble predictions could lead to more accurate and reliable results. By combining the strengths of various machine learning algorithms, such as support vector machines, decision trees, or neural networks, we could create a more comprehensive and powerful predictive framework. Moreover, implementing a gender classification filter could further refine our predictions by filtering out irrelevant or biased data, enhancing the model's ability to accurately discern facial expressions and emotions across different genders. With more time and resources dedicated to these enhancements, our facial and emotion detection models can achieve even greater accuracy and efficacy. Additionally, incorporating color image training into our model can lead to significant improvements in performance. Color images contain richer visual information compared to grayscale images, enabling the model to capture more detailed features and nuances in facial appearances. By training the model on color images, we can enhance its sensitivity to color variations and improve its overall accuracy in facial recognition tasks. In the end, we can conclude that with a better dataset and better methods, our models would perform better. Lastly, by employing a tactic of ensemble modeling we are sure we could get far better results.