Knock Knock, Who's There: Facial Recognition using CNN-based Classifiers

Qiyu Sun, Alexander Redei
Department of Computer Science
Central Michigan University
Mount Pleasant, MI

***Abstract*—Artificial intelligence (AI) has captured the public's imagination. Performance gains in computing hardware, and the ubiquity of data have enabled new innovations in the field. In 2014, Facebook's DeepFace AI took the facial recognition industry by storm with its splendid performance on image recognition. While newer models exist, DeepFace was the first to achieve near-human level performance. To better understand how this breakthrough performance was achieved, we developed our own facial image detection models. In this paper, we developed and evaluated six Convolutional Neural Net (CNN) models inspired by the DeepFace architecture to explore facial feature identification. This research made use of the You Tube Faces (YTF) dataset which included 621,126 images consisting of 1,595 identities. Three models leveraged pretrained layers from VGG16 and InceptionResNetV2, whereas the other three did not. Our best model achieved a 84.6% accuracy on the test dataset.**

***Keywords*—Face recognition; deep learning; convolutional neural networks; DeepFace**

I. Introduction

Facial recognition is a method of identifying an individual using his or her face from a digital image or a video clip. Such methods could be used for facial authentication by pinpointing and determining facial features from a given image, uniquely identifying the person. Initially this was limited to desktop computers due to demanding computational power constraints. Recently however it has seen wider usage, such as on mobile devices, robotics, finding missing people, and diagnosing diseases. Facial recognition is also applied in diagnosing diseases. 22q11.2 deletion syndrome (22q11.2 DS) is the most common micro-deletion syndrome and was underdiagnosed in a variety of populations in the past. Because the disease results in multiple defects throughout the body, including cleft palate, heart defects, a characteristic facial appearance, and learning problems, healthcare providers often can't pinpoint the disease, especially in diverse populations. After analyzing the disease with facial analysis technology, researchers found that sensitivity and specificity were greater than 96% for all populations, which demonstrated how facial analysis technology can assist clinicians in making accurate 22q11.2 DS diagnoses [1]. Researchers with the National Human Genome Research Institute (NHGRI), part of the National Institutes of Health, and their collaborators, have successfully used facial recognition software to diagnose a rare, genetic disease in Africans, Asians, and Latin Americans [2].

Facial recognition technology has a wide range of applications and profound social and cultural impacts and has been introduced across various aspects of public life. For example, facial recognition payment services are now possible. Nowadays, in China people can purchase food at the grocery store and can even complete their payment directly by scanning their face at a register without needing a credit card or mobile application [3]. In terms of the design and implementation of security systems, facial recognition technology also has a wide range of applications including web and mobile authentication [4], airport check-in [5], and smart medicine cabinets [6]. In the education industry, facial recognition has been applied to compulsory schooling to address issues such as campus security, automated registration, and student emotion detection and has largely been seen as routine additions to school systems with already extensive cultures of monitoring and surveillance [7]. While facially driven learning has been widely used, critical commentators are beginning to question the pedagogical limitations of it. They purposed multiple questions about facial recognition technology including the likelihood of it altering the nature of schools and schooling along divisive, authoritarian and oppressive lines, and what kind of law and regulatory mechanisms can help for eliminating the potential risks to consumers when they are making use of it [7]. Due to the relatively limited technology, the current ability to detect human faces in this field provides a buffer from coping with the potential consequences including a serious threat to online identities being misused by hackers for illegal activities.

An overview of the rest of the paper is as follows: in Section 2 we reviewed some of the related work in the same research field with DeepFace; in Section 3, we introduce core techniques related to DeepFace: Deep Learning and Convolutional Neural Networks; Section 4 describes the 3D model-based face alignment method applied and the model architecture used; Section 5 talks about our deep learning model that follows the architecture of DeepFace's and was trained on YouTube Faces (YTF) video data set. In Section 6 we present some quantitative results of our models and the last section is the conclusion.

II. Related Work

Biometric facial recognition, also known as automatic face recognition, is a particularly attractive method of biometric recognition because it focuses on "faces," the same identifiers that humans primarily use to distinguish people. One of its main goals is the understanding of the complex human visual system and the knowledge of how humans represent faces in order to discriminate different identities with high accuracy. Facial recognition consists of three basic processes: detection, capture, and face match. The detection process is to determine

if there is a target face in the source. The capture process transforms the targeted face into a set of digital data based on the facial features. The face match process verifies if the two faces are of the same person. This process is shown in Fig. 1 below.

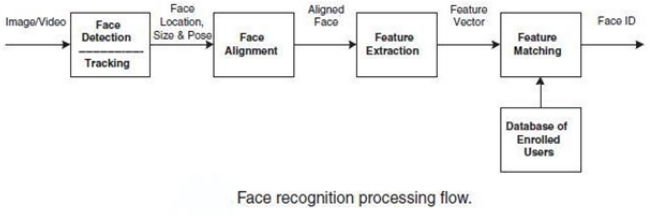

Fig. 1. Facial Recognition Processing Flow.

A large number of approaches have emerged in the field of facial recognition, including a hand-crafted features based method [8] and a widely applied metric learning methods with task-specific objectives [9]. These approaches were never quite able to reach human-level performance in identifying faces. Although progress in facial recognition was encouraging, the task has also turned out to be a difficult endeavor.

A. DeepFace

DeepFace is a deep learning face recognition technology developed by a research group at Facebook. It identifies human faces in digital images with human-level performance. In DeepFace, researcher revisited both the alignment step and the representation step of the face recognition process and proposed a new approach of deriving a face representation by employing explicit 3D face modeling. It employed a nine-layer neural net with over 120 million connection weights and was trained on four million images uploaded by Facebook users [10] [11]. DeepFace demonstrates that a 3D model-based alignment method can effectively help in face recognition and closes the gap to human-level accuracy. Next Generation Identification (NGI) is another application developed by Federal Bureau of Intelligence (FBI) of the same year. According to one report the NGI's performance is non-satisfactory. It returns a ranked list of 50 possibilities and only promises an 85% chance of returning the suspect's name in the list [12]. The DeepFace system (stated by the Facebook Research team) reaches an accuracy of 97.35 ± 0.25% on labeled faces in the wild (LFW) data set whereas human beings have 97.53% [13]. Google FaceNet later achieved a 99.65% accuracy on the same data set [14].

B. Local Binary Patterns (LBP)

Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision and it is the particular case of the Texture Spectrum model proposed in 1990 [15]. LBP was first described in 1994 [16] and it is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number [17]. Using the LBP combined with histograms we can represent the face images with a simple data vector [18]. In the LBP approach for texture classification, the occurrences of the LBP codes in an image are collected into a histogram. The classification is then performed by computing simple histogram similarities. However, considering a similar approach for facial image representation results in a loss of spatial information and therefore one should codify the texture information while retaining also their locations. One way to achieve this goal is to use the LBP texture descriptors to build several local descriptions of the face and combine them into a global description. The basic methodology for LBP based face description proposed by Ahonen et al. [19] is as follows: The facial image is divided into local regions and LBP texture descriptors are extracted from each region independently. The descriptors are then concatenated to form a global description of the face, as shown in Fig. 2.

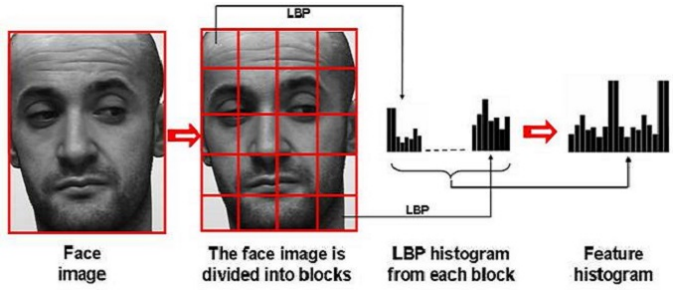

Fig. 2. Face Description with Local Binary Patterns.

C. DeepID-Net

A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem similar to the one that we want to solve. As for most image detection problems, the main features of the objects to be detected are often similar, so a pre-trained model can be leveraged to typically get improved performance. But researchers found a gap between the pre-training task and the fine-tuning task that makes pre-training less effective [20]. Inspired by the need to adopt more targeted optimization solutions for specific objects, researchers propose the DeepID-Net model. DeepID-Net is an image detection model developed by the Multimedia Laboratory of the Chinese University of Hong Kong. Its full name is DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection [20]. They added more steps on the region-based convolutional neural networks on the region-based convolutional neural networks (R-CNN) processing, including bounding box rejection, deep model training, def pooling layer, SVM-net(replace softmax with hinge loss to accelerate learning), multi-stage training, etc, as shown in Fig. 3. The model yields a 99.8% accuracy, while the state-of-the-art method achieves a 97% accuracy when testing multi-view facial images [20]. This paper was published on CVPR2014. After that, the team focused on applying the model to the specific application of facial recognition, and correspondingly made some changes and optimizations to the DeepID model. The updated two versions of the model are called DeepID2 and DeepID3.

D. FaceNet

FaceNet is a universal system that can be used for face verification (is it the same person?), recognition (who is this person?) and clustering (looking for similar people?) [22]. The

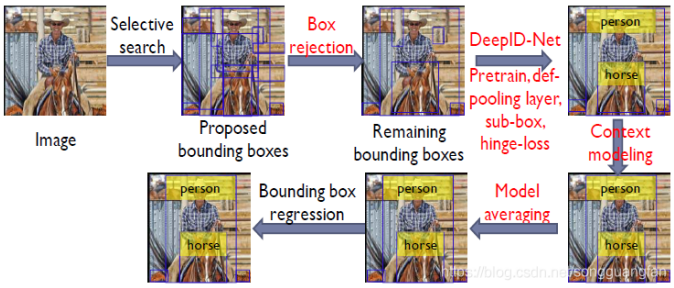

Fig. 3. Overview of DeepID-Net Process (Texts in Red Highlight the Steps that are Not Present in RCNN.) [21]

method adopted by FaceNet is to map images into Euclidean space through a convolutional neural network. Different from the application of other deep learning methods on human faces, FaceNet did not use the traditional softmax method to classify and learn, and then extract a certain layer as a feature. It directly used triplets-based LMNN (Maximum Boundary Neighbor Classification) loss function to train the neural network, and the network directly outputs a 128-dimensional vector space [22]. FaceNet has achieved an accuracy of 99.63 ± 0.09% on the LFW dataset and an accuracy of 95.12 ± 0.39% on the YTF dataset [22]. The advantage of this model is that the target image can be used with very little processing. It also provides future research directions, such as analyzing wrong samples to improve accuracy, reducing model size to speed up training, etc.

III. Methodology

Deep learning is a specific subfield of machine learning [23]. It represents learning process from data, emphasizing on learning successive ”layers” of increasingly meaningful representations. The word ”deep” in ”deep learning” is not referring to deeper understanding achieved through the approach but stands for the idea of successive layers of representations. The number of layers that contribute to the model is called the depth of the model. These layered representations are learned through models called neural networks and they are structured in layers stacked one after the other. Deep learning is technically a mathematical framework for learning representations from data with a multi-stage way. A large deep network has multiple layers with many more nodes in each layer, which leads to many more parameters to tune. It would be too slow and insufficient to train a deep learning model without a large dataset and powerful computers. Compared to the traditional learning algorithms (Regression, Random Forest, Support Vector Machine, etc.), deep learning may not necessarily outperforms when given data of small scale. But once the data scale goes up exponentially, deep learning outperforms others because more parameters provide the capability to learn complicated nonlinear patterns [24]. Generally, we expect the model to capture the most helpful features by itself without too much expert-involved manual intervening on features learning.

Machine learning is about mapping inputs to target outputs. The specification of what each layer does to their input data is stored in a bunch of parameters called ”weight”. The learning process refers to finding a set of values of the weights of all layers in a network so that the network will correctly map inputs to their associated targets. But here comes the issue: to find the correct value for all of the weights can be a daunting task, especially when modifying the value of one parameter will affect the performance of the whole model. To control the output of a neural network, the loss function plays an important role in making the prediction of the network and the target. It computes a distance score measuring how well the network performs. The job of the ”optimizer” is to use this score as a feedback signal to adjust the value of the weights, successively trying to lower the loss score. Implementing ”back-propagation” is the central algorithm used for this in deep learning architectures. In recent years, deep learning has achieved a revolution with tremendous achievements on many types of difficult problems, especially perceptual problems, which have long been historically difficult for machine learning.

Convolutional neural networks are a type of feed-forward artificial neural networks, most commonly applied to analyzing visual imagery [25]. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics [26] [27]. They have applications in image classification, Image segmentation, character recognition [28], medical image analysis [29], natural language processing [30].

A convolutional neural network consists of an input layer, multiple hidden layers and an output layer. In any feed-forward neural network, all middle layers are called hidden layers due to their inputs and outputs are sealed by the activation function and final convolution [31]. Convolution refers to a mathematical operation between two matrices, it is defined as the integral of the product of the two functions after one is reversed and shifted. It then evaluates the integral over all values of the shift to produce a convolutional function. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers [32]. The convolutional layer has a defined fixed small matrix, also called a kernel or filter. It computes the element-wise multiplication of the values in the kernel matrix and the original image values as the kernel is sliding, or convolving, across the matrix representation of the input image as shown in Fig. 4. Specially designed kernels can fast and efficiently process images for common purposes like edge detection and many others. Convolutional and pooling layers respond to feature extraction. The fundamental difference between a densely connected layer and a convolution layer is that dense layers learn global patterns in their input feature space, while convolution layers learn local patterns [33].

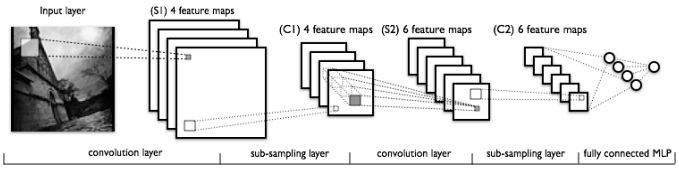

Fig. 4. The LeNet Architecture Consists of Two Sets of Convolutional, Activation, and Pooling Layers, Followed by a Fully-connected Layer, Activation, Another Fully-Connected Layer, and Finally a Softmax Classifier [34].

We used the development environment provided by Google Colab with the TensorFlow and Keras stacks. All the models

were trained using NVIDIA Tesla T4 graphical processing units (GPU) hardware equipped with 16 GB of memory and 12.7 GB RAM. This allowed us to create and simulate a deepface-esq model architecture.

For the interested reader, a link to our source code repository can be found here: https://github.com/QiyuSun/Facial-Recognition-using-CNN-classifier. This repository includes our Jupiter notebook python files as well as a readme file with links to the YTF dataset. The full dataset is not included in our repo due to space constraints, we only link to it.

IV. DeepFace Architecture

Before the deepface architecture came about, one of the challenges to facial recognition was the reduced accuracy caused by face images collected from different perspectives. In fact, facial alignment is still considered a difficult issue, especially in an unsupervised environment. The task of face alignment is to automatically locate key facial feature points, such as eyes, nose tip, mouth corners, eyebrows, and contour points of various parts of the facial contour according to the input face image. The process of face alignment can be divided into three sub-problems: 1) How to model the apparent image (input) of a human face? 2) How to model the face shape (output)? 3) How to establish the association between the apparent image (model) of the face and the shape (model) of the face? In terms of the DeepFace method, Facebook researchers have made a great contribution in the development of an effective deep neural network architecture with a very large, labeled dataset of faces, an effective facial alignment system based on explicit 3D modeling of faces, and results that reach near real time human-level performance [13].

A. Alignment Pipeline

The alignment pipeline of DeepFace is as follows:

(a) Detect face with 6 initial points.
(b) Crop out the face with 2D-aligned inducing.
(c) Apply Delaunay triangulation by 67 fiducial points on the 2D-aligned crop, adding triangles on the contour to avoid discontinuities.
(d) Transform triangulated face into 3D shape.
(e) The face becomes a deep 3D triangle net.
(f) Deflect the triangulation.
(g) The final frontalized crop.
(h) A new view generated by the 3D model (not used in paper).

The function of these steps uses the 3D model to align the face, so that the CNN can exert its maximum effect. This is shown in Fig. 5.

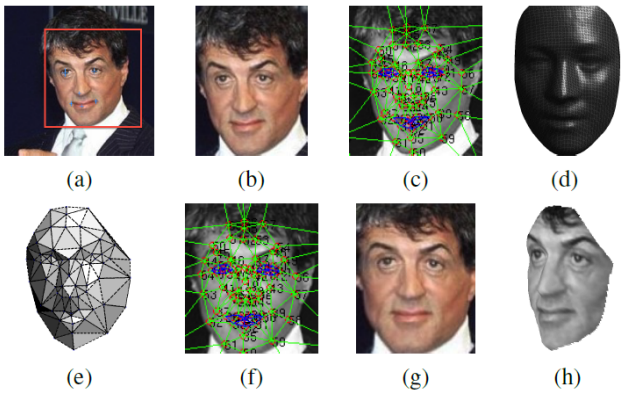

Fig. 5. DeepFace Alignment Pipeline.

B. Representation

After 3D alignment, the images formed are all shrunk into 152×152 pixel inputs into the network structure shown in Fig. 6, the parameters of the structure are as follows:

(a) The 3D aligned 152x152 pixel 3-channel RGB face image is sent to the convolutional layer (C1) with 32x11×11×3 filters.
(b) Then the obtained 32 feature maps are fed to the maximum pooling layer (M2), and the 3×3 spatial neighborhood is maximum pooled with a stride of 2. Each channel is executed separately.
(c) After M2 is a convolutional layer with 16x9×9×16 filters (C3).
(d) Locally connected l [35], but each position in the feature map learns a different filter bank. Local means the parameters of the convolution kernel do not share. Locally connected layer is different from the convolutional layer in its kernel.
(e) F7 and F8 are fully connected layers and they can capture the correlation between the features of the face image, such as the position and shape of the eyes and mouth. The output of F7 will be used as the original face representation feature vector with 4096 dimensions. The face representation on F8 is sent to K-way Softmax to generate the probability distribution on the category label for classification. The 4030 dimension is respective to the number of identities in the SFC training data set, and each identity has 800 to 1200 face pictures.
(f) Normalization of face representation: Normalize the face representation feature to be between zero and one to reduce the sensitivity to changes in illumination.

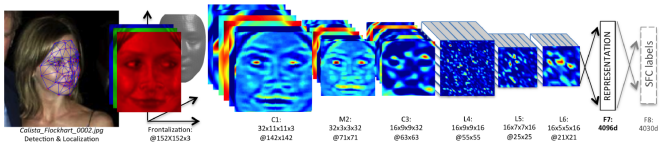

Fig. 6. Outline of the DeepFace Architecture.

C. Datasets

In DeepFace, researchers trained their model on the Social Face Classification (SFC) dataset and evaluated the model

on the Labeled Faces in the Wild database (LFW) and the YouTube Faces (YTF) dataset. In this work, our model is built on the YTF dataset. The YTF Database is a face video database, which aims to study the problem of unconstrained face recognition in videos. It contains 3,425 videos of 1,595 different identities. All the videos were downloaded from YouTube. It provides an average of 2.15 videos available for each subject with clips duration varying from 48 frames to 6,070 frames, and 181.3 frames of average length. It initially performs automatic screening to ensure that the videos are long enough to capture useful information for the various recognition algorithms with stable detection. The remaining videos were manually verified to ensure that the videos would be correctly labeled corresponding to the subjects, not static images or slides, and no duplicated videos were included [36].

In terms of designing the data set structure and benchmarks, the YTF dataset follows the principal of the Labeled Faces in the Wild (LFW) collection. All video frames are encoded with well-built descriptors with the face detector output considered in each frame. The face images are bounded and cropped from the frame, 2.2 times of their original sizes. Additionally, the images are resized to 200x200 pixels then cropped into 100x100 pixels in central area. The images are aligned by fixing the coordinates of facial feature points following a conversion to grayscale. The image is divided to a fixed grid of blocks with the descriptions of each block normalized to a unit Euclidean length [36]. For the benchmark tests of the YTF dataset, the YTF dataset follows the example of the LFW benchmark various tests like standard test and ten-fold test. It is divided into 5,000 video pairs and 10 groups, for evaluating video-level face verification [36].

V. Model Training and Evaluation Protocol

Six models were built and trained. First we built a baseline model, next a frame base model, aligned base model, VGG16 base model, InceptionResNetV2 base model, and finally InceptionResNetV2 model. The training and validation distribution for all models followed the same split.

A. Dataset

Twenty images were extracted for the train set of each identity. The remaining images in folders of each identity are divided into training set and validation set with ratio of 8:2. All base models were built on a subset of YTF frame_images_DB and aligned_image_DB with 160 classes, which consisted of the first 10 percent of videos ordered by name. It is of note that in practical face recognition applications today, the images processed by the model are often already aligned.For example in our dataset, each image is assigned a unique floating point number corresponding to it's identity. For example Figure 7 corresponds to the unique identifier '0.614', where the '0' indicates the folder with the identity of a known actor — Aaron Eckhart — and '614' indicates the particular frame sequence in the dataset. Because pre-aligned images were already available, we did not implement an alignment subsystem. Instead after implementing a model architecture, we trained and validated that model on the aligned_image_DB dataset.

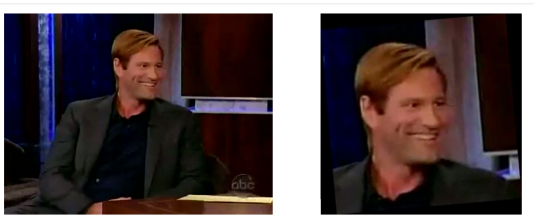

Fig. 7. '0.614' Images of Aaron Eckhart in Frame_Image_DB (left) and Aligned_Image_DB (Right).

B. Baseline Model

Our first approach initially applied a CNN-based model with a single Conv2D layer to train a baseline model on frame_image_DB subset for developing a better performing model. The baseline model consisted only one Conv2D layer with 32 nodes, input shape of (152, 152, 3) and activation function relu. The output layers consisted of 160 nodes and the pooling window sizes were 3 by 3 and 2 by 2 for Conv2D and MaxPooling2D, respectively. After converting the pooled feature map to a single column, only one Dense layer was defined and which also served as output layer using softmax for multi-class classification. Categorical_crossentropy, Adam, and accuracy were defined in compiling for loss, optimizer and metrics. After training, the accuracy of training and validation reached 99.84% and 97.64% within 20 epochs Fig. 8.

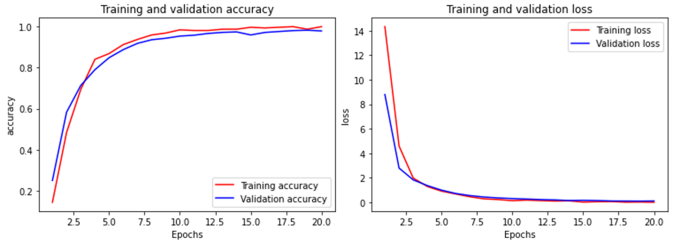

Fig. 8. Baseline Model Performance.

C. Frame and Aligned Base Model

After the baseline model was developed, we added one more Conv2D layer with 16 nodes and relu as activation function, one more dense layer with 1024 nodes to train the Frame Base Model and Aligned Base Model, essentially structuring our model to the model architecture of DeepFace. Since the data set is relatively large, and there was no obvious overfitting observed during the training process, Regularization methods were not added. The learning rate of optimizer Adam was 0.00002. Both Frame Base Model and Aligned Base Model were trained to 20 epochs. The frame base model reached 97.23% and 95.15% accuracy of training and validation Fig. 9. The Aligned Base Model reached 86.51% and 80.65% accuracy of training and validation Fig. 10.

D. VGG16 and InceptionResNetV2 Base Model

The next iteration constructed utilized the VGG16 Base Model and InceptionResNetV2 Base Model following a similar architecture. The Conv2D layers and MaxPooling layers of both two models were replaced with pre-trained conv_base.

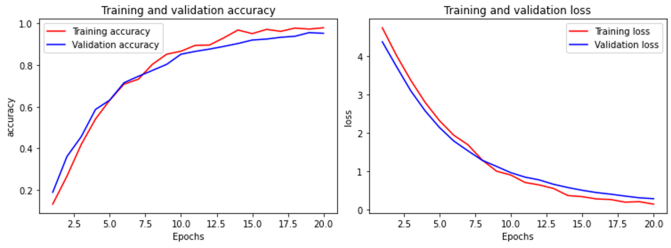

Fig. 9. Frame Base Model Performance.

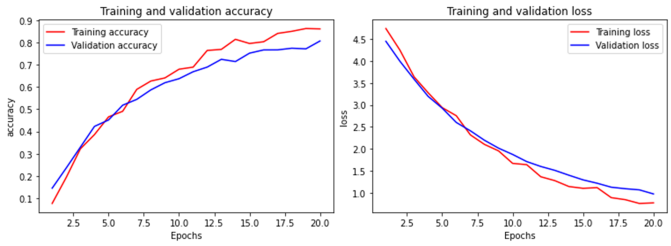

Fig. 10. Aligned Base Model Performance.

We froze the top layers of conv_base so that weights of those layers would keep unchanged during training process. The first dense layer was set 4096 nodes. After training, the VGG16 Base Model reached 98.76% and 97.03% accuracy of training and validation Fig. 11. The InceptionResNetV2 Base Model reached 99.75% and 97.23% accuracy of training and validation Fig. 12.

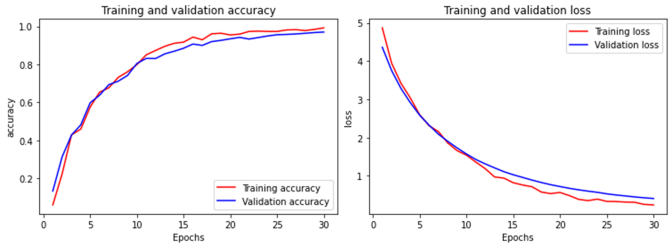

Fig. 11. VGG16 Base Model Performance.

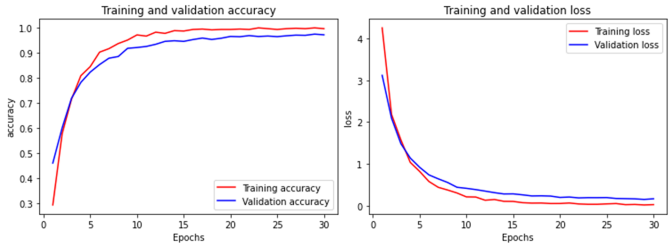

Fig. 12. InceptionResNetV2 Base Model Performance.

E. InceptionResNetV2 Model

The final InceptionResNetV2 Model shared the same model architecture with the InceptionResNetV2 Base Model. It was trained on the entire aligned_image_DB dataset (621,126 images of 1,595 identities) with same dataset distribution as the base dataset. Aftering training with 200 epochs, the InceptionResNetV2 model reached 91.12% and 90.79% accuracy of training and validation as shown in Fig. 13.

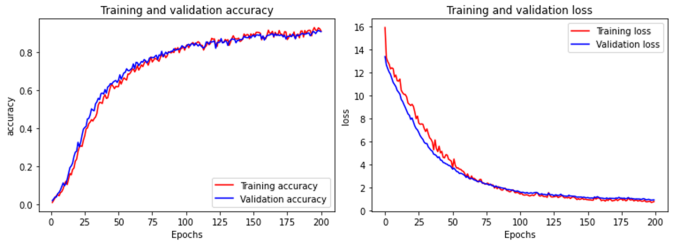

Fig. 13. InceptionResNetV2 Model Performance.

VI. Results

The basic baseline model showed great performance in accuracy (99.84% on train set, 97.64% on validation set, 98.60% on test set) but it was just a simple classifier that allowed us to explore what parameters needed to be tweaked to obtain even better results. We can use the performance of other models as a baseline to evaluate the performance of the all models trained on the specific dataset. Based on the baseline models, we developed the Frame Base Model and Aligned Base Model. The Frame Base Model reached 97.23%, 95.15% and 90.78% accuracy on train set, validation set and test set. The Aligned Base Model reached 86.51%, 80.65% and 64.92% accuracy on train set, validation set and test set (Table I, Table II). The Frame Base Model considerably outperformed but the result was not that convincing. It was built on different datasets with the same architecture, which implied that the difference in performance could be caused by the specific dataset used for validation. At this point, we believe the Base Model learned little from facial features and instead was emphasizing on other factors too much. It was those no-facial related features that helped it reached greater performance than the Aligned Base Model, meaning it could be objects like chairs, studio backgrounds, etc.

The VGG16 Base Model and the InceptionResNetV2 Base Model were trained on subset of aligned image subset with more neurons added in dense layers and more training epochs compared to previous two models. The VGG16 Base Model reached 94.98% and 93.47% of train and validation accuracy at the 20th epoch, and the InceptionResNetV2 Base Model reached 99.63% and 96.58% of accuracy, respectively. After 30 epochs of training, the VGG16 Base Model reached 98.76%, 97.03% and 89.60% on train set, validation set and test set (Table I, Table II), and the InceptionResNetV2 Base Model reached 99.75%, 97.23% and 93.96% of accuracy on train set, validation set and test set (Table I, Table II). The accuracy curves of the two models were both smooth, which indicated that no apparent overfitting was observed in training. On the ground, we can see the great performance of pre-trained model layers in image classification.

The finalized InceptionResNetV2 Model was built with the same structure as the InceptionResNetV2 Base Model. It had 1,595 neurons in the final output layer corresponding to the number of indentities in the completed aligned image dataset (621,126 images of 1,595 identities) and was trained

with 50 epochs. It took 0.033s for each image during the training process. The model reached accuracy of 92.75% and 91.46% at the 198th epochs then it performed increasingly higher loss and relatively lower accuracy afterwards (Table III). The decline in performance may be caused by a variety of factors, including poor architecture of model and overfitting. Considering that no explicit overfitting was found in previous models, it would be of help to promote the model performance with a better networks architecture rather than with additional regularization methods added. In the DeepFace architecture, researchers added three locally-connected layers after 3D convolutional layers and maxpooling layers, which might be one of the solutions for structural optimization of the model. In the final evaluation on test set, the InceptionResNetV2 Model performed 84.60% of accuracy and 1.2582 of loss, which demonstrated that more tuning on the model were required, as well as some pre-operations on images before being fed into model.

TABLE I. PERFORMANCE OF TRAINING & VALIDATION ACCURACY

Model	Train-Acc	Val-Acc
Baseline Model	99.84%	97.64%
Frame Base Model	97.23%	95.15%
Aligned Base Model	86.51%	80.65%
VGG16 Base Model	98.76%	97.63%
InceptionResNetV2 Base Model	99.75%	97.23%
InceptionResNetV2	91.12%	90.79%

TABLE II. PERFORMANCE ON TEST DATA

Model	Test-Loss	Test-Accuracy
Baseline Model	0.0866	98.60%
Frame Base Model	0.6357	90.78%
Aligned Base Model	1.8023	64.92%
VGG16 Base Model	0.9026	89.60%
InceptionResNetV2 Base Model	0.2992	93.96%
InceptionResNetV2	1.258	84.60%

TABLE III. INCEPTIONRESNETV2 MODEL PERFORMANCE IN FINAL EPOCHS

Epoch	Loss	Accuracy	Val_loss	Val_Accuracy
195	0.7641	0.9294	0.9466	0.9062
196	0.7965	0.9156	0.9768	0.8975
197	0.8055	0.9131	0.9380	0.9082
198	0.7046	0.9275	0.9005	0.9146
199	0.7391	0.9231	0.9204	0.9106
200	0.7834	0.9112	0.9276	0.9079

We were not able to best DeepFace's 96% accuracy. However, our top model achieved a respectable 84.6%. Considering the constrained resources we had (this was developed entirely on a single laptop and Google Colab compared to the massive resources available at Facebook), the mission of this project was achieved.

We directly used the aligned dataset published in YouTube Face dataset rather than implementing a specific face alignment method. In DeepFace, the method developed to map 2D human facial features to 3D models and use them as 3D input to train models was key to making DeepFace achieve its breakthrough outstanding performance. In addition, unlike our training and verification based entirely on the YouTube Face (YTF) dataset, DeepFace's training set and verification set involved a total of three different data sets (Social Face Classification dataset, Labeled Faces in the Wild dataset, and YTF dataset). Training on one dataset and using the different datasets for validation reduces its accuracy deviation when facing images of different sizes and types. When looking only at the YTF dataset making full use of these factors, DeepFace achieved the test accuracy of 91.4 ± 1.1%. This number is a more accurate threshold to compare our model against as it's an apples-to-apples comparison leveraging the same dataset. Taking into account the limitations of so many conditions mentioned above, the result we obtained, when compared to DeepFace, seems to be rewarding.

Although we were inspired by the architecture of DeepFace, as described in detail above we did not fully copy the DeepFace model. We were limited by the computational power available to us. Just the memory required to fully reproduce the DeepFace model is massive and greatly exceeds that which we had access to. Instead, we demonstrated that the simpler and more accessible models we built have promise in recreating DeepFace-style breakthrough performance, utilizing a fraction of the resources.

VII. CONCLUSION

DeepFace revolutionized the facial image recognition industry. In this paper, we demonstrated the power of learned features through six convolutional neural networks (CNNs). Inspired by the DeepFace architecture, but in making our own tweaks, the models we constructed were trained on the YouTube Faces (YTF) dataset to be multi-class classifiers. The Base Models showed satisfactory performance, which indicated that a CNN-based architecture could manifest remarkable performance in image classification if given a large dataset. Compared to the DeepFace model architecture, the model was less complex, which potentially would bring difficulties in capturing essential facial features effectively in other datasets. Based on the first experiment, it was obvious that there are multiple factors that needed to be taken into consideration when constructing an image classifier for a face recognition system. When dealing with exponentially massive amounts of data, the architecture and depth of the model will play a crucial role in performance. The success of DeepFace showed that remarkable results could be achieved with the right architecture combined with face alignment and frontalization. We demonstrated that our models could obtain good results at much lower computational cost.

The first few models we built showed us many factors that need to be considered when building a large CNN classifier, such as: how to make full use of the structure and characteristics of CNN itself, the suitable combination of hyperparameters for training, and how to adjust particular parts of model architecture when working with datasets at large scales. This paper is a valuable contribution to the field of image classification and facial recognition.

In future work, we wish to further improve the model. First, we would explore adding some preprocessing methods for face images, such as image sharpening, extended face alignment, and frontalization. Second, in terms of the model architecture, we may consider trying to combine layer functionalities such as with locally-connected layers or pooling layers.

ACKNOWLEDGMENT

This work was funded by the Michigan Aerospace Center for Simulations. We are grateful for our colleagues and the support of Central Michigan University.

REFERENCES

[1] B. F. M. Cuneo, "22q11.2 deletion syndrome: Digeorge, velocardiofacial, and conotruncal anomaly face syndromes," *Current Opinion in Pediatrics*, vol. 13, 2001.

[2] P. Kruszka, Y. A. Addissie, D. E. McGinn, A. R. Porras, E. Biggs, and M. Share. . . , "22q11.2 deletion syndrome in diverse populations," in *American Journal of Medical Genetics Part A, 2017; 173 (4): 879 DOI: 10.1002/ajmg.a.38199*.

[3] Y. li Liu, W. Yan, and B. Hu, "Resistance to facial recognition payment in china: The influence of privacy-related factors," *Telecommunications Policy*, vol. 45, no. 5, p. 102155, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0308596121000598

[4] I. Olade, H.-n. Liang, and C. Fleming, "A review of multimodal facial biometric authentication methods in mobile devices and their application in head mounted displays," in *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2018, pp. 1997–2004.

[5] T. Zhu and L. Wang, "Feasibility study of a new security verification process based on face recognition technology at airport," *Journal of Physics: Conference Series*, vol. 1510, no. 1, p. 012025, mar 2020. [Online]. Available: https://doi.org/10.1088/1742-6596/1510/1/012025

[6] S. Yamanaka and V. Moshnyaga, "New method for medical intake detection by kinect," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2018, pp. 218–221.

[7] M. Andrejevic and N. Selwyn, "Facial recognition technology in schools: critical questions and concerns," *Learning, Media and Technology*, vol. 45, no. 2, pp. 115–128, 2020. [Online]. Available: https://doi.org/10.1080/17439884.2020.1686014

[8] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[9] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *BMVC*, 2013.

[10] T. Simonite, "Facebook creates software that matches faces almost as well as you do," *MIT Technology Review*, 2014.

[11] C. NEWS, "Facebook's deepface shows serious facial recognition skills," *CBS NEWS*, 2014.

[12] R. Brandom, "Why facebook is beating the fbi at facial recognition," *The Verge*, 2014.

[13] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[14] S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2020, pp. 1–5.

[15] L. Wang and D.-C. He, "Texture classification using texture spectrum," *Pattern Recognition*, vol. 23, no. 8, pp. 905–910, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0031320390901358

[16] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, 1994, pp. 582–585 vol.1.

[17] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 32–39.

[18] K. S. do Prado, "Face recognition: Understanding lbph algorithm," *towards datascience*, 2017.

[19] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[20] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, and X. Tang, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] songguangfan, "A detailed explanation of deepid-net," June 2020.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends® in Machine Learning*, 2009.

[24] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *In ANIPS*, 2012.

[25] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev, and N. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378475420301580

[26] W. Zhang, K. Doi, M. L. Giger, Y. W. R. M. Nishikawa, and R. A. Schmidt, "Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network," *Medical Physics*, vol. 21, pp. 517–524, 1994.

[27] W. Zhang, K. Itoh, J. Tanida, and Y. Ichioka, "Parallel distributed processing model with local space-invariant interconnections and its optical architecture," *Appl. Opt.*, vol. 29, no. 32, pp. 4790–4797, Nov 1990. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-29-32-4790

[28] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[29] B. Kayalibay, G. Jensen, and P. V. D. Smagt, "Cnn-based segmentation of medical imaging data," *ArXiv*, vol. abs/1701.03056, 2017.

[30] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[31] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[33] A. Hue, "Dense or convolutional neural network part 1 — architecture, geometry, performance," *Medium*, 2020.

[34] L. Weng, "An overview of deep learning for curious people," June 2017.

[35] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2518–2525.

[36] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR 2011*, 2011, pp. 529–534.