ID++ : Integrating YOLO v3 with FaceNet Embeddings for Identity Recognition and Demographic Classification

KEH, Sedrick Scott OR, Ka Po UY, Mark Christopher
The Hong Kong University of Science and Technology
{sskeh,kpor,mcsuy}@connect.ust.hk

Abstract

In this report, we propose a novel hybrid architecture for real-time facial detection, recognition, and classification. We use the state-of-the-art YOLO v3 detection algorithm, combined with feature extraction using FaceNet embeddings. By replacing FaceNet's custom facial detection algorithm with YOLO v3, we hope to increase the speed of recognition while simultaneously maintaining the accuracy of FaceNet. Empirical trials reveal a significant improvement in real-time recognition, both in terms of speed and accuracy, reaching an average speed of 5 frames per second. Furthermore, multiple classifiers were experimented upon for the final layer, which does the classification for identity, age, gender, and race. Quantitative evaluation suggests using a combination of KNNs, SVMs, multilayer perceptrons, and wide residual networks in order to optimize the classifier performance.

1. Introduction

Facial recognition is a field with a wide variety of applications, ranging from social media filters and automatic photo tagging to advanced security systems and authentication for buildings. This may also be used in the educational setting, such as in taking attendance for lectures or exams, greatly reducing the effort required to perform such menial tasks.

A major challenge of facial recognition systems such as these is the wide variability in the faces, as well as the large number of classes in identification systems. Additionally, there is a question of how to best mathematically represent a face in a way that makes computation both efficient and accurate. Meanwhile, in age, gender and race classification, there are very few fixed features that can be successfully and consistently used for identification. For instance, a person who is 40 years old may look very similar to another who is 30 years old even though there is a 10-year age gap between the two people.

Furthermore, a large portion of state-of-the-art systems make use of static photos for recognition and classification. On the contrary, this report tackles the challenge of real-time data, which is more difficult in terms of both efficiency and accuracy. This is due to a greater emphasis on the speed of classification, as well as the need to identify multiple moving faces at once, including those that constantly enter and exit the screen. This choice to use real-time video data is motivated by its potential uses in security and authentication systems.

2. Datasets and Features

For this project, we used two different publicly available datasets of faces, namely Labeled Faces in the Wild (LFW) dataset [3, 4] and the UTKFace dataset [13].

2.1. LFW Dataset

The Labeled Faces in the Wild dataset contains over 13,000 photos of various celebrities labeled with the appropriate names. In this dataset, 1680 identities have 2 or more distinct faces. These were used in training the FaceNet model to learn the appropriate feature vectors.

2.2. UTKFace Dataset

The UTKFace Dataset contained over 23,000 unique faces, labelled with the age, gender, and eth-

nicity. In the dataset, age ranged from 0 to 116, gender was either male (0) or female (1), and ethinicity was numbered 0 (White), 1 (Black), 2 (Asian), 3 (Indian), 4 (Others). These faces were used to trian the classifiers for the age, gender, and ethnicity prediction.

3. Methods

3.1. System Architecture

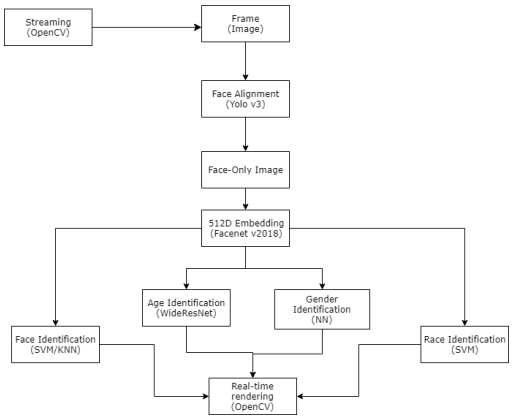

Figure 1. Real time Face Analysis System Architecture.

The task of facial recognition and classification can be broadly broken down into three parts, namely the face detection from video data (which uses YOLO v3), getting embeddings (which uses FaceNet), and classification (which uses a variety of classifiers). Classification can further be broken down into identity classification and age/gender/race classification. These steps will be detailed further in the succeeding sections.

3.2. YOLO v3

The original YOLO v3 model [8] was trained using the Common Objects in Context (COCO) training dataset. This dataset has 330,000 images, 1.5 million object instances, and 80 object categories. The YOLO v3 model is the third installation in the YOLO-series of object detection models [7], and provides some improvements over predecessors, such as more convectional layers for their neural network architecture, called Darknet 53. The model we used [6] also poses impressive results on the COCO test dataset, as seen in the image below.

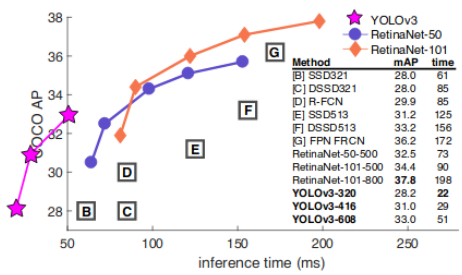

Figure 2. Results on different models on the COCO testdev dataset (MaP).

3.3. FaceNet

FaceNet [10] is a deep learning model that directly learns a 160x160-to-512D mapping from a huge collection of human face images for estimating the difference and similarity of face images in an Euclidean space.

Figure 3. FaceNet structure. FaceNet consists of a batch input layer and deep CNNs followed by L2 normalization, which results in the face embedding. This is followed by the triplet loss during training[10].

FaceNet is trained using deep convolution neural networks with 7.5 millions of hyper-parameters and a triplet loss function, achieving the state-of-the-art accuracy of 99.6% on the Labeled Faces in the Wild dataset [3, 4]. The triplet loss aims to minimize the Euclidean distance between the embeddings of a anchor image and a positive image, as well as maximizing the Euclidean distance of a anchor image and a negative image.

Our model used is based on a FaceNet model pre-trained by David Sandberg [9]. His model was trained on the VGGFace2 (3.31 million images of 9131 subjects) [2] and it achieved a accuracy of 99.65% on the LFW dataset. With the use of transfer learning, we trained his pre-trained Tensorflow model with our targeted faces, which helps to focus on the features of our targets. Face images in the bounding boxes generated by the YOLO v3 model in each frame are cropped and then passed to the FaceNet for a 512-dimensional embedding of the related human face image.

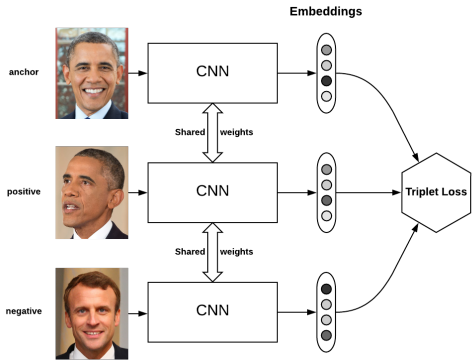

Figure 4. Triplet loss on two positive faces (Obama) and one negative face (Macron)[5].

3.4. Identity Classification

For the identity classification task, there are 5 classes to identify to: (”Bingyi JING”, ”David OR”, ”Sedrick KEH”, ”Mark UY”, ”Unknown”). For each of the classes, around 40 photos were taken. These were then cropped and aligned using YOLO to prepare them for classifiation. A sample of these faces is shown below.

Figure 5. Training dataset for identity classification. There are 5 classes (4 known faces and 1 unknown).

The first step was to convert the training data of faces into FaceNet embeddings. Once we gathered the FaceNet embeddings with the respective classes, we passed these to various classifiers. The classifier outputs probabilities from a softmax function, and it considers an image as a match to an existing identity if the output probability is greater than 0.60, as shown below.

$$pred_y = \begin{cases} \arg\max f(x) & \max f(x) > 0.50 \\ \text{Unknown} & \max f(x) \leq 0.50 \end{cases} \quad (1)$$

Empirical experiments using our own faces suggested that SVMs were the models that performed the best, followed by KNNs. As such, SVMs was the final choice for this identity classification model.

3.5. Age, Gender, Race Classification

For the age, gender, and race classification, we first got feature vectors from the input faces using FaceNet embeddings. We used the faces from the UTKFace dataset to train three separate models for predicting age, gender, and race. These faces were first converted to 512 dimensional feature vectors using FaceNet. Furthermore, the labels were extracted to match with the corresponding faces. The trained classifier thus takes in a 512 dimensional vector as input and outputs a prediction.

For the age, we used a Wide Residual Network [11, 12], a deep CNN-based model, trained specifically to predict ages from given human faces. For genders, we used a Multilayer Perceptron Network, and for race, we used SVMs. These were experimented upon empirically and were shown to be the optimal combination of models to use in terms of accuracy and robustness.

4. Discussion

We successfully integrated two well-trained and state-of-the-art deep learning models with several classifiers to create ID++. With the help of OpenCV library [1], ID++ is able to stream video input from either a computer web camera or an IP camera in real time. In practice, due to the limited computation power, we only achieved real-time face analysis (25 frame-per-second) for 352 x 240 (240p) resolution video streaming but this has already shown the possibility of distributed real time face analysis system with the same architecture.

However, we also discover the following challenges:

1. Current prediction is based on one frame only. We think the prediction on frames should have dependency on previous and next frames to

achieve object tracking and to recognize back of people.

2. The embedding is not very sensitive towards non-white face. We think it is because the pre-trained FaceNet is trained on mainly white faces and we should train on a relatively diverse and balanced dataset in terms of the skin colors.

3. Efficient and robust classification is still under development. Our SVMs and KNNs classifers cannot classifier the "UNKNOWN" label well when there is a face image of a person that is not in the database. We should try more different classification models and use image processing technique to transform images taken under different lightings.

Figure 6. Screen-shot of the real time system.

5. Conclusion and Future Work

Although a lot was accomplished in this paper, there still remains a lot to be done to improve ID++. As of now, ID++ is simply a collation of different models, such as YoloV3, FaceNet, and various classification models. However, not much time was spent to try and fully integrate all the different models together. One potential room for improvement is to train the FaceNet model on the images of known identities in the local database. This method uses the idea of transfer learning to create more representative feature vectors of a person's face. This is a very powerful idea, which can greatly boost the accuracy of the identification task, and removes the need for an extra classification layer at the end of the model. This task is quite simple, but the lack of computation resources made it impossible to partially retrain on the FaceNet model/ architecture. As a result, a SVM model was used. Other potential tasks for future work is to try different models, both deep learning based, and SVM based to help boost the identification performance, which is actually the most difficult portion of the model. There are also more optimization steps, which can be done to help improve the speed of the entire network from detection to classification. As mentioned, a lot of work remains to be done, however, the current results prove to be very promising.

6. Acknowledgements

We thank Professor JING, Bing-Yi for his instruction and introducing the area of computer vision for facial recognition.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[3] G. B. Huang, M. Ramesh, T. Berg, and rik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. (07-49), October 2007.

[4] G. B. H. E. Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. (UM-CS-2014-003), May 2014.

[5] O. Moindrot. Triplet loss and online triplet mining in tensorflow. `https://omoindrot.github.io/triplet-loss`, 2018.

[6] T. Nguyen. Deep learning-based face detection using the yolov3 algorithm. `https://github.com/sthanhng/yoloface`, 2019.

[7] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[8] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[9] D. Sandberg. Face recognition using tensorflow. https://github.com/davidsandberg/facenet, 2018.

[10] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[11] S. Zagoruyko and N. Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[12] C. Zhang. Easy real time gender age prediction from webcam video with keras. https://github.com/Tony607/Keras_age_gender, 2018.

[13] S. Y. Zhang, Zhifei and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.