Video-based Face Recognition

Mukhamejan Karatayev
Nazarbayev University
Nur-Sultan, Kazakhstan
mukhamejan.karatayev@nu.edu.kz

Assylan Akimbayev
Nazarbayev University
Nur-Sultan, Kazakhstan
assylan.akimbayev@nu.edu.kz

Almukambet Abduokhapov
Nazarbayev University
Nur-Sultan, Kazakhstan
almukambet.abduokhapov@nu.edu.kz

Abstract

Nowadays, face recognition is one of the most actively studied problems in computer vision and biometrics. Especially, video-based face recognition offers a plethora of potential applications in the real-world including visual surveillance, access control, and video content analysis. In this work, a face recognition system secured with a liveness detection model was implemented. The results suggest that the proposed network can accurately detect faces from a video stream, identify whether it is fake or not, and recognize the corresponding person.

1. Introduction

Driven by key law authorization and commercial applications, investigation on confront acknowledgment from video sources has heightened in recent times. The following has illustrated that recordings have special properties that permit both people and mechanized frameworks to perform acknowledgment precisely in troublesome seeing conditions. Be that as it may, critical inquiries about challenges stay as most video-based applications don't permit for controlled recordings.Face recognition in unconstrained situations remains challenging for most commonsense applications. In contrast to conventional still-image based approaches, as of late the exploration about the center has moved towards video-based approaches.

In this work, a smart video-based face recognition system was implemented. The key feature of this project is that the proposed algorithm also includes a Convolutional Neural Network (CNN) based liveness detection system. Which secures our face recognition model from being easily fooled by "spoofing" and "non-real" faces. The video recording is divided into frames, then a pretrained Caffe face detector was utilized to obtain coordinates of faces at each frame. If the detected face is recognized by liveness detection as the real one, a 128-d embedding is generated for it. And, these embeddings were used to recognize the faces of the characters in the video stream. The architecture of each component will be discussed in later sections in more detail.

The remainder of this paper is organized as follows: Section II reviews different works related to this field. Section III describes the dataset utilized in this work. In Section IV we will introduce the methodology of this work, Section V defines methods that were used to evaluate proposed models. While, Section VI reports our experiments and describes our results. Finally, in the conclusion section, key achievements of this project were outlined and we will discuss the issues and propose suggestions for the future work.

2. Literature Review

There are three main modules required for a face recognition system: face detection, face alignment, and face recognition. There are a variety of different approaches for each of the modules. Therefore, different architectures were analyzed and compared to find the most suitable one.

The paper published by Zhang et al. [10] proposed a deep cascaded multi-task framework for face detection and alignment. It was used in our project due to its great performance. The overall framework integrates these two tasks with the implementation of unified cascaded CNNs by multi-task learning. Firstly, it identifies the candidate windows using shallow CNN. Then, with the more complex CNN structure, the non-face windows are dropped. At the end, an even more powerful CNN layer refines the windows from previous stages to produce facial landmark positions. The main advantage of this approach is the appropriate size of the network for real time performance. There are also other methods for face detection, for example Yang et al.

[5] proposed deep convolutional neural networks for facial attribute recognition. However, it requires much more time for inference because of the highly complex network structure. While, for face alignment there are regression-based models [8] and template fitting methods [9] widely used. However, they ignore the inherent correlation between face detection and alignment tasks. Also, they are limited by handcrafted features and show considerably worse performance compared to cascaded multi-task framework [10]. For example, on a benchmark dataset, RCPR [8] and TSPM [9] had a mean squared error equal to 11.6 and 15.9 respectively. The model proposed by Zhang et al. had a MSE equal to 6.9, so heavily outperforming previous approaches.

Due to the extraordinary success of CNN models on ImageNet challenge, for the last module of face recognition systems the typical CNN architectures such as AlexNet, GoogleNet, VGGNet, SENnet and ResNet are mostly used. Also, some assembled networks were introduced, Hu et al. [2] found that accumulating outputs of assembled models increases the recognition performance compared to individual networks. However, the 34-layer residual nets proposed by He et al. [3] were used for face recognition purposes. As compared to plain networks such as VGGNet [4], residual networks have lower complexity and better performance. For example, ResNet34 has 3.6 billion FLOPs (multiply-adds) while VGG-19 network has about 19.6 billion FLOPs. Also, according to tests on the same datasets, residual solutions have significantly better accuracy compared to plain models [3].

In the past years, numerous well-established strategies have been proposed to confront acknowledgment issues in various domains. In A. Yilmaz [11] the two-dimensional appearance of the confront picture is treated as a vector by filtering the picture in lexicographical order, with the vector measurement being the number of pixels within the picture. In the Eigen – face approach S. Li [12], all confront pictures consist of an unmistakable confront subspace. This subspace is direct and crossed by the eigenvectors. But PCA does not accomplish exactness in terms of acknowledgment since the creation of the confront subspace does not altogether separate between people. The transient demonstration based strategies learn the transient, facial elements of the confrontation all through a video. In this method Z. Kal [13] performs Hidden Markov Models (HMM) which employs a picture preparation library by forcing movement data on it to prepare a Well. It probabilistically generalizes a still – picture library to do video-to-video coordinating. Preparing these models is computationally costly, particularly when the dataset estimate is expansive. S. Oron [14] employs all accessible data, to recognize the cast instead of the facial data alone. It employs a complex for known characters which effectively cluster input outlines. P. Viola and M. Jones[15] formulate tracking as a probability density propagation problem and the algorithm provides verification results. However, no systematic evaluation of recognition was done. The major challenge with this approach is to find such a region of interest.

In the literature on face retrieval, numerous strategies have been proposed [16], [17], [18]. Sivic et al. [19] proposed a strategy that recovers the target subject employing a set of pictures that contains broad varieties of models. Arandjelovic et al. [20] proposed an end to-end video confrontation recovery framework with a few handling steps. There are also many works on face tracking and association. Zhou et al. [21] joined appearance-based models in a molecule channel to realize vigorous visual following. While Roth et al. [22] proposed a multi-pose-conflict following approach in two stages utilizing different prompts. Comaschi et al. [23] also proposed a web multi-face tracker utilizing locator certainty and an organized SVM. An efficient tracker, SORT [24], accomplishes comparable to other state-of-the-art strategies with 20 times quicker speed. Du et al. [25] proposed a conditional random field (CRF) framework to associate faces by utilizing the similarity of facial appearance, location, motion, and body appearance.

3. Dataset

The dataset contains 3,425 videos of 1,595 different people. An average of 2.15 videos are available for each subject.The shortest clip duration is 48 frames, the longest clip is 6,070 frames, and the average length of a video clip is 181.3 frames [7]. Figure 1 represents the example frames from the YouTube Faces data set. While, Figure 2 shows the distribution of videos per person.

Figure 1: Example frames in the Dataset

The bottom row depicts some of the challenges of this set, including amateur photography, occlusions, problematic lighting, pose, and motion blur.

All video outlines are encoded using several well-established, face-image descriptors. Specifically, we consider the confront detector and after a change to grayscale, the images are adjusted by settling the arrangements of automatically detected facial features. For facial recognition

#videos	1	2	3	4	5	6
#people	591	471	307	167	51	8

Figure 2: A number of videos per person

networks,we will use a 128-d feature vector for output that is consumed to quantify the face.

4. Methodology

As already mentioned in the previous section, face recognition consists of several tasks, which are compiled into a pipeline. The first and most important task of the pipeline is detecting faces in the given frame. Then, these faces go through a liveness detection model, which discards all faces that are categorized as "fake" (spoofed). In the third step, from remaining faces feature vector is created. At last, these feature vectors are compared with already known persons' face features to recognize the name of the person.

Face detection Face detection is a crucial part of the model. Therefore the most accurate face detector must be implemented. We have chosen CNN based face detection as its performance, in terms of accuracy, is better than other types of detection. However, such performance comes with a cost of longer time to process each frame. Thus, a GPU is required for this task.

The approach we employed for face detection is a 4-stages structure model that uses CNN at the core. These stages are shown in the Figure 3.

As the first stage, the pre-processing stage takes the image and resizes it into several images of different scales. In the second stage, the model finds candidate patches bounded by the box, which could be a face. Using these bounding boxes and the non maximum suppression technique, the number of bounding boxes is reduced by removing candidates which are definitely not faces and merging overlapping bounding boxes [10].

In the third stage, similar to the second stage, the model employs a more detailed CNN to get more refined candidates of bounding boxes. At last, again it uses much more detailed CNN to get the bounding box and find 5 key points, namely eyes, tip of the nose, edges of the mouth, inside the bounding box [10]. The structure of the network is shown in the Figure 4.

Liveness detection After detecting faces, we need to make sure that the detected face is real, meaning it is an image of the face directly taken from the webcam or security cameras. As face recognition has become one of the most used unlock features on the phones, some security measures

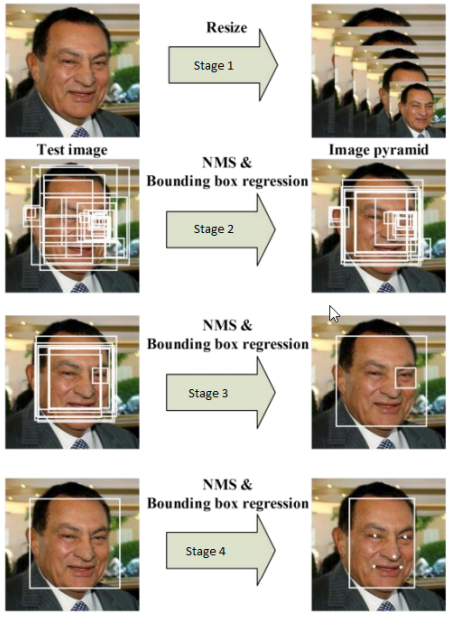


Figure 3: Stages of Face Detection

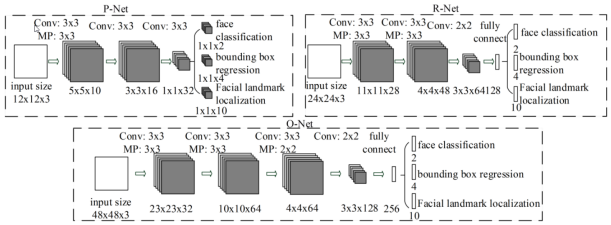


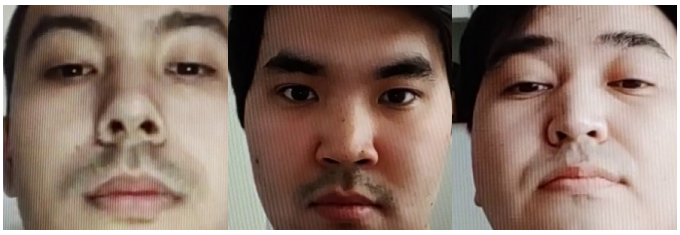
Figure 4: Convolutional Neural Network

must be implemented. That is, say, if someone recorded the face of a certain person and showed this recording to unlock this person's phone (spoofing).

To detect if the face is real or fake, we again employed CNN model, which consists of two CNN layers. While liveness detection model is basic CNN, it works fast due to this simplicity. To train this model, we have recorded 3-8 seconds videos of ourselves placing them in the "real" folder, then re-recorded these videos but from the screen of the monitor. Obtained videos were placed in a "fake" folder. Later, all faces from these videos are extracted in places in respective folders for training of the model.

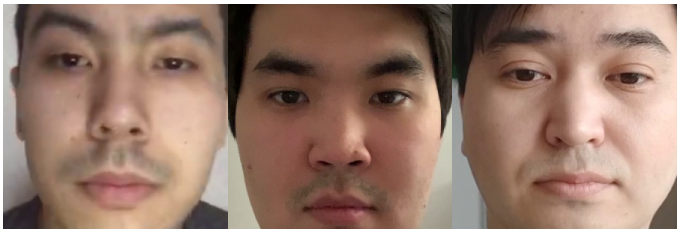
The dataset for liveliness model, we obtained 155 real faces and 174 fake faces. The dataset was split into training and testing samples with 75% and 25% ratio, respectively. The model was trained for 100 epochs with an initial learning rate of 0.00001, and batch size was set to 8.

From the Figure 7, it can be seen that towards the end of training cycles, the model has a very high accuracy score of 0.99. This model is then used to detect if the face is real or not.



(a) (b) (c)

Figure 5: Examples of Fake faces.



(a) (b) (c)

Figure 6: Examples of Real faces.

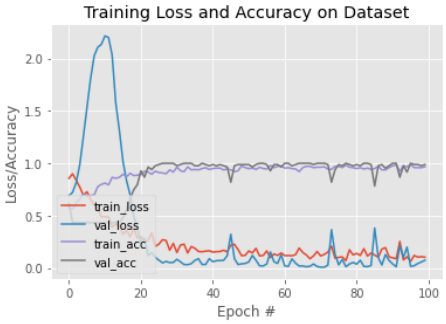


Figure 7: Training of liveness detection model.

Face Recognition For face recognition model, we are planning to use a pre-trained model that is a ResNet model.

With deeper networks, the problem of vanishing gradient arises, however, to deal with the problem, ResNet was developed. ResNet introduces the notion of skips. Say, there are layers *A1* and *A2* in the network. The *A1* is connected to *A2* with activation function **g**. In a plain network, not ResNet, input of *A2* is

$$g(W \times A1 + B),$$

where W is vector of weights and B is a bias [3]. However in ResNet, input of A2 is

$$g(W \times A1 + B + A1),$$

thus A1 skips forward a layer [3].

The model that we are going to use is based on ResNet34, but with fewer layers, and the number of filters reduced by half.

This model was trained on 3 million images of Labeled Faces in the Wild dataset and reaching the accuracy of 99.38%. In general terms, this model extracts a 128 dimensional vector of face features (embedding). We will have to run all our images in the database and store the array of features in either memory or locally. Then, features from the input image will be compared with features from the images stored locally using Knn+votes to recognize the face. Thus, resulting label is the one that has most votes.

In other words, after detecting the face was real, we extract features from the face and compare it with features of faces that are stored in the database by taking the distance measurement between two vectors. If this distance is less than threshold (in our case 0.6), then the detected face is considered to belong to the face with the least distance.

Preliminary procedures In order to start recognizing faces, we first need to create a database of faces that we would like to recognize. All faces that were not recognized are classified as "UNKNOWN". Therefore, we have again recorded a short video of ourselves to extract faces and placed them in the database folder. Each person's videos are stored in the folder named after that person, which was done for convenience.

All faces are then converted into feature vectors and stored in the dictionary.Thus, updating this database is relatively simple. After we have created a database of known faces, we then can do face recognition.

In our model, we did not perform face tracking, rather we took each frame as a separate entity and detected faces found in the frame. After all faces are detected, we iteratively perform face recognition on each face.

5. Evaluation Metric

In order to evaluate whether our face recognition model is successful, evaluation metrics should be defined first. There are different types of metrics depending on the real-life problem for which model is used, such as face verification, open-set or close-set face identification. However, in each task we have a set of known subjects stored somewhere, and given a new subject to be compared with this set. Face verification is actually a one-to-one comparison between a known set and new object to find whether two people are the same. While, for open-set and close-set problems, one-to-many comparisons are required to identify the identity of a given subject.

One of the most common metrics is accuracy. It repre-

sents the percentage of correctly classified pairs

$$Acc = \frac{CorrectlyClassifiedPairs}{TotalComparedPairs} \times 100\%$$

Modern FR algorithms commonly have accuracy around 99% on benchmark datasets.

The general definition for the Average Precision (AP) is calculating the area under the precision-recall curve. As the precision and recall are always between 0 and 1, therefore AP is bounded within 0 and 1. The exact calculation of AP may vary depending on the dataset, therefore special libraries for such calculations were used.

There are also some special metrics for face recognition problems. The receiver operating characteristic (ROC) measures true accept rate (TAR) - the fraction of unique comparisons that exceed the threshold, and the false accept rate (FAR) - the fraction of impostor comparisons that falsely exceed the threshold. Usually, it is important to keep FAR very low, as FR methods are mostly used for security reasons.

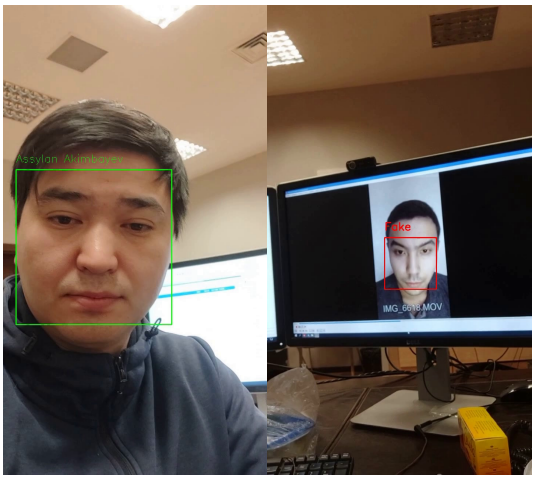
6. Results & Analysis

It is needless to say that this model's performance is great at detecting faces, and recognizing the faces. In addition, given good lighting conditions and better resolution video, then liveness detection works well. Liveness detection still has room to improve, which is discussed in the Future Works section. By varying the threshold for face recognition, we can either be more strict or less strict depending on application. Say, if we are using this model for authentication with face recognition, then we should use stricter thresholds as the cost for false positives are very high in these circumstances.

6.1. Error Analysis

In the experiments, two settings were used. First if the detected face is real, then it is bounded by a GREEN bounding box. In this case, the model will try to identify the face with known faces, if there was not a face it could identify as, the label *UNKNOWN* in red will appear, otherwise the name of the person it identified with will appear in green color. If it is detected as spoofed, then it is bound by the RED box without initiating face recognition. In the second settings, we let the model recognize faces even if the detected face is spoofed. The difference can be seen in the images below Figure 8.

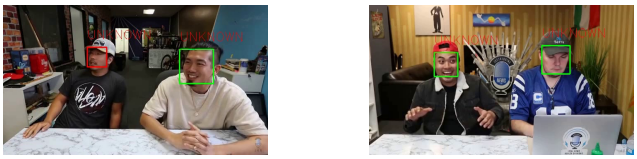
Several experiments were conducted, in one of the videos, we recorded a real face switching to a face on the monitor (Figures 8a, 8b). Model successfully detects both a real face and a spoofed face. However, when turning the camera angle covering half of the real face, it detects it as a spoofed face and cannot identify the face (Figures 8c, 8d).



(a) (b)

(c) (d)

Figure 8: The output of experiment 1.



(a) (b)

Figure 9: The output of experiment 2.

In Figure 8c, where the known face is not recognized was due to a known faces sample. Known faces are extracted from the frontal images, therefore it does not deal well with images where some parts of the face is covered.In the future, this can be addressed by implementing augmentation of the images in the known faces processing part.

In the Figure 9 below, we can see that the model detects

two faces, however one is detected as spoofed, another one is detected as real. We are assuming that this is also due to training of the liveliness model. In the liveliness model, we trained on images of ourselves in a fairly lit environment. Therefore, we think that the model thinks that if a face is somewhat darker, then it is a spoofed face (Figure 9a).

The same video, but from another angle in Figure 9b. The faces on the people are lighter, therefore they are being detected as real. In addition, since these people are not in the known faces database, they are labeled as UNKNOWN.

6.2. Future Work

In the future, liveness detection needs to be improved by training on images of different faces, different environment settings. As of right now, since it was trained on relatively few training samples, it does not do well in low light environments.

References

[1] Chakraborty, S., Das, D. (2014). An overview of face liveness detection. arXiv preprint arXiv:1405.2227.

[2] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales (2015), "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," in ICCV workshops, pp. 142–150.

[3] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[4] K. Simonyan and A. Zisserman (2015). Very deep convolutional networks for large-scale image recognition. In ICLR.

[5] S. Yang, P. Luo, C. C. Loy, and X. Tang (2015), "From facial parts responses to face detection: A deep learning approach," in IEEE International Conference on Computer Vision, pp. 3676-3684.

[6] Wang, M., Deng, W. (2018). Deep face recognition: A survey. arXiv preprint arXiv:1804.06655.

[7] Wolf, L., Hassner, T., Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In CVPR 2011 (pp.529-534). IEEE.

[8] X. P. Burgos-Artizzu, P. Perona, and P. Dollar (2013), "Robust face landmark estimation under occlusion," in IEEE International Conference on Computer Vision, pp. 1513-1520.

[9] X. Zhu, and D. Ramanan (2012), "Face detection, pose estimation, and landmark localization in the wild," in IEEE Conference on Computer Vision and Pattern Recognition, pp. 2879-2886.

[10] Zhang, K., Zhang, Z., Li, Z., Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499-1503.

[11] Yilmaz, O. Javed and M. Shah, Object Tracking: A Survey, ACM Computing Surveys, vol. 38, no. 4, June (2006).

[12] S. Li and Z. Zhang, Float Boost Learning and Statistical Face Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1112–1123, November (2004).

[13] Z. Kalal, J. Matas and K. Mikolajczyk, Pn Learning: Bootstrapping Binary Classifiers by Structural Constraints, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 49–56, April (2010).

[14] S. Oron, A. Bar-Hillel, D. Levi and S. Avidan, Locally Order Less Tracking, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1940–1947, January (2012).

[15] P. Viola and M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, no. 2, pp. 511–518, August (2005).

[16] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa, "Triplet probabilistic embedding for face verification and clustering," CoRR, vol. abs/1604.05417, 2016.

[17] E. G. Ortiz, A. Wright, and M. Shah, "Face recognition in movie trailers via mean sequence sparse representation-based classification," in CVPR, 2013, pp. 3531–3538.

[18] R. G. Cinbis, J. J. Verbeek, and C. Schmid, "Unsupervised metric learning for face identification in tv video," ICCV, pp. 1559–1566, 2011.

[19] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: Video shot retrieval for face sets," in CIVR, 2005.

[20] O. Arandjelovic and A. Zisserman, "Automatic face recognition for film character retrieval in feature-length films," in CVPR, 2005, pp. 860–867. 19

[21] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," in IEEE TIP, 11 2004, p. 14911506.

[22] M. Roth, M. Bauml, R. Nevatia, and R. Stiefelhagen, "Robust ¨ multi-pose face tracking by multi-stage tracklet association," ICPR, 2012.

[23] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal, "Online multiface detection and tracking using detector confidence and structured svms," 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6, 2015.

[24] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in ICIP, 2016, pp. 3464–3468.

[25] M. Du and R. Chellappa, "Face association for videos using conditional random fields and max-margin markov networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 9, pp. 1762–1773, Sep. 2016.