CNN Based Key Frame Extraction for Face in Video Recognition

Xuan Qi, Chen Liu, Stephanie Schuckers
Clarkson University
8 Clarkson Ave, Potsdam, New York
{qix, cliu, sschucke}@clarkson.edu

Abstract

Nowadays we see an increasing demand for face in video recognition. However, in order to overcome the large variations in face quality in video streams, as well as for the purpose of improving the processing speed of face recognition system, frame selection becomes a necessary and essential step prior to performing face recognition. In this paper, we propose a convolutional neural network (CNN) based key-frame extraction (KFE) engine with Graphic Processing Unit (GPU) acceleration, which targets at extracting key-frames with high quality faces correctly and swiftly. We evaluated our method with ChokePoint dataset following NIST standards and compared against several representative key-frame selection approaches. The experimental results show that our CNN-based KFE engine can largely reduce the total processing time for face in video recognition, as well as improves the recognition accuracy of the face recognition back-end. With GPU acceleration, our KFE engine reaches and exceeds real-time processing speed requirement under HD resolution, making it capable of processing multiple video steams on the fly. On top of that, our proposed KFE engine is adaptive to different face recognition back-end.

1. Introduction

Face in video recognition is widely used in video surveillance and video analytics. In recent years, various face recognition (FR) engines [18, 14, 17] have been proposed and achieved very high accuracy on still-image based datasets. In real world scenarios, however, applications such as live-video surveillance and large-volume video footage processing pose great challenges on face recognition. On the time constraint side, for surveillance purpose we need real-time processing speed to meet the frame rate of video cameras at 24 to 30 frames per second (FPS). For playback purpose, we also need to process large video repositories by the required deadline, especially in a criminal investigation or other law enforcement applications. On the resource limitation side, first, face detection and recognition requires a lot of computation resources, especially when every frame needs to be processed; second, transferring continuous video streams adds a huge burden on the network traffic and high demand on the network bandwidth.

To overcome the challenges of time constraint and resource limitation, one promising strategy is to perform key-frame selection on video streams. A key-frame is the frame which can represent the content of the scene [22]. In this work, we propose a convolutional neural network (CNN) based key-frame extraction (KFE) engine that extracts the key-frame according to the face quality, which benefits the face recognition by reducing the data volume and providing key-frames with high quality faces.

Our CNN model is trained in such a way that it will generate a predicted score that indicates the quality of the face in the frame. The selection of the key frame is based on this score without the need to perform the face recognition. Then, the selected key frames are sent to a deep neural network (DNN) based back-end for face recognition. With the proposed design, our target is to improve the face recognition accuracy of the overall face in video recognition system, reduce the data volume transferred over the network, as well as improve the real-time processing capability of the face in video recognition system.

The rest of paper is organized as follows: In Section 2 we explain the need for key frame extraction, as well as provide an overview of related works. Our CNN-based key-frame extraction (KFE) engine design is introduced in Section 3. In Section 4, we designed several experiments to validate our KFE engine design through detailed analysis on the accuracy and performance of face recognition system. Finally, we conclude our work in Section 5.

2018 IEEE 4^{th} International Conference on Identity, Security, and Behavior Analysis (ISBA) 978-1-5386-2248-3/18/$31.00 ©*2018 IEEE*

2. Background

2.1. The need for key frame extraction

Frame selection is an essential step for modern face in video recognition system, because performing face recognition on every frame is computationally expensive and normally is not required for common application scenarios. Although more frames can provide more information, sometimes poor quality information will actually lower the face recognition performance. The typical reasons for getting poor quality faces are: faces with bad lighting condition or non-frontal faces detected by the face detector, and non-face images erroneously extracted by face detector. In Figure 1, we use a video sequence of one identity from Choke-point dataset [20] as an example. The largest sub-image on the right is a good quality face which is clear and of good illumination condition, detected from a stream of video frames. But, from the other sub-images of the same identity, we can see that some faces detected from video frames can be small sized or poorly illuminated. What's more, in second row, we can even see some face detection errors which are not faces. These poor quality faces or detection errors can lower the face recognition performance, causing false positive or false negative results. For example, a face of poor quality can increase the chance of being recognized as somebody else, or the chance of being not recognized as someone in the gallery, against the fact the person to whom the face belongs is in the gallery.

2.2. Related Works

The existing frame selection methods can be divided into three main categories [3]: clustering based, optical flow based and quality based.

The clustering approach considers a video as a set of face images and transfer them into a high dimensional space formed by feature vectors. The key-frame is decided by clustering algorithms such as K-means [5, 11]. The optical approach selects key-frames according to inter-frame motion extracted by optical flow algorithm such as Lucas-Kanade [7, 16].

For qualify based approach, Nasrollahi et al. introduced four metrics: face symmetry, sharpness, contrast and brightness to evaluate a face image quality for frame selection [12]. Anantharajah et al. also applied a similar metrics-based quality assessment system for face clustering in news video [1]. The face quality metrics based approach comes with a low computation overhead, which makes high-speed real-time processing possible. For instance, Qi et al. achieved faster than real-time ($\geq$30FPS) processing speed on full HD videos (1920×1080) by using metrics based face quality assessment [15]. But, all works [12, 1, 15] mentioned above need pre-defined empirical weights to be associated with different quality metrics in order to form a final quality score. It would be difficult, however, for the fixed weights to be adaptive to different videos under different scenes. As a result, Chen et al. [2] proposed a learning-to-rank based method to make face quality assessment adaptive to different face recognition methods. They used three databases: high quality face images from controlled environment, face images from uncontrolled environment, and non-face images for training all feature weights. Then, all weighted feature vectors: Histogram of Oriented Gradients (HOG), Garbor, Gist, Local Binary Patterns (LBP) and CNN are combined to one rank-based quality score (RQS) vector. Finally, the RQS vector is transformed into a unique quality score by using a linear function. Another work proposed in [8] also followed the learning-to-rank framework, in which the mismatch between training and test images is considered as another factor along with the visual quality of face image. The overall accuracy is improved compared with the work of Chen et al. [2]. Vignesh et al. proposed a deep CNN based face quality assessment for face in video selection [19]. By utilizing the good feature learning capability of DNN, they outperformed the rank-based approach [2] without extracting pre-defined facial features.

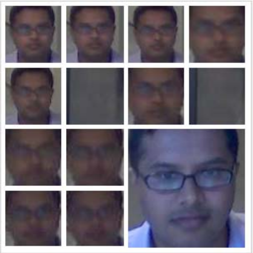

Figure 1: Examples of poor quality faces extract from video frames

Our scheme falls into the quality-based category. The main differences between our work and previous works mentioned above are the following. Firstly, our approach does not apply the quality evaluation method of the entire image. Instead, we concentrate on the face quality evaluation to guide our KFE engine. Secondly, we do not use predefined empirical weights or knowledge to evaluate face quality. Instead, when constructing the model of our KFE engine, we utilize the information from FR back-end during the training process, which is essential for our face quality evaluation. Thirdly, in addition to addressing the face recognition accuracy, we take performance in practical usage into consideration and employ metrics such as frame processing rate and data volume reduction to evaluate our scheme. Lastly, to meet the real-time processing speed, Graphics Processing Unit (GPU) acceleration is applied in all stages of our KFE engine, i.e., face detection and key-frame analysis at front-end and face recognition in back-end.

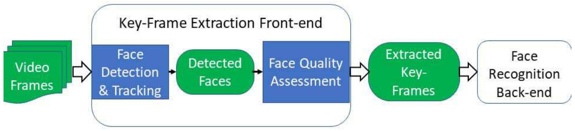
Key-Frame Extraction Front-end
Video Frames
Face Detection & Tracking
Detected Faces
Face Quality Assessment
Extracted Key-Frames
Face Recognition Back-end

Figure 2: Face recognition with key-frame extraction

3. CNN Based Key-Frame Extraction

In this section we present our CNN based key frame extraction engine design, where we employ a quality based frame selection strategy.

3.1. Face quality based key-frame extraction

A conventional face recognition system is composed of three main processing procedures: face detection, feature extraction and face recognition [21]. Once a face gets detected, the system will extract the feature vectors like LBP and HOG of the face images. Next, the extracted feature vectors are encoded and classified in face recognition module which generates the verification or identification results.

Our overall system architecture is illustrated in Fig.2. We add a face quality assessment (FQA) module after the face detection and tracking module, and group them as the key-frame extraction (KFE) front-end of our system. The FQA module will evaluate the quality of all detected face images and store the frame with best quality face image as “key frame”. Instead of transferring all the frames, now only key frames are forwarded to the back-end to perform face recognition. As a result, the KFE can effectively reduce the data volume and processing overhead of the FR back-end.

The detailed design of our face tracking and key-frame flowchart is shown in Fig. 3. At the beginning, we apply Viola Jones HAAR feature based cascade classifier for face detection [10]. Please note, the face detection procedure here is fully parallelized and boosted with GPU acceleration. Here, all detected faces are tracked according to their locations in the scene. Next, if the number of detected faces does not change, we deem it is the same scene and perform the face quality evaluation on detected faces. If the face quality evaluation shows that current face image is better, this frame will be used as key frame of the corresponding face. This is applied to every face in the frame. If the number of faces does change, we deem there is a change in the scene and output the current key-frames of all the faces to the back-end and clear the key-frame buffer. Then, we will start another round of key frame evaluation.

3.2. CNN based face quality assessment

As the core of our KFE engine, we apply CNN based face quality assessment (FQA) module to evaluate the quality of all detected face images and extract key frames based on the evaluation results accordingly. Different from just giving an absolute score to a face image [12, 1, 2], our FQA module is designed to utilize the information from FR back-end, based on which to make the key frame selection decision. As shown in Fig. 4, the core of the FQA module is a convolutional neural network (CNN) that will generate predicted recognition performance score of the input face image. During the training phase of CNN, first, every face image in the training set is sent to FR back-end to obtain a recognition performance score scaled between 0 to 100%. Next, the face image associates with its real FR performance score are forwarded to attend the training of CNN. During the testing phase, the trained CNN is used to predict the FR performance score of incoming new face image, which is defined as the face quality in this work. The entire training and inference procedures of our CNN model are all accelerated with CAFFE [6] framework with GPU acceleration.

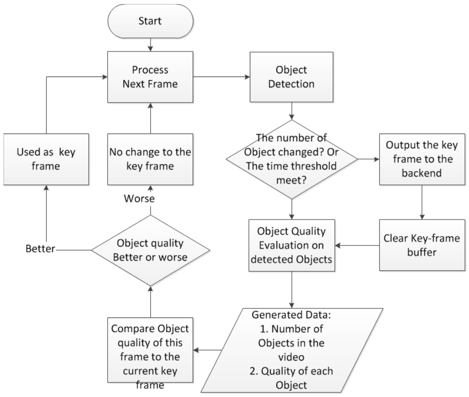
Start
Process Next Frame
Object Detection
Used as key frame
No change to the key frame
The number of Object changed? Or The time threshold meet?
Output the key frame to the backend
Worse
Better
Object quality Better or worse
Object Quality Evaluation on detected Objects
Clear Key-frame buffer
Compare Object quality of this frame to the current key frame
Generated Data:
1. Number of Objects in the video
2. Quality of each Object

Figure 3: Face tracking and key-frame extracting flowchart

We designed our CNN with the structure describe in Fig. 4, based on image-based regression method. Different from the work by Vignesh et al. [19], we increased the complexity in convolution layers in order to extract more features. We also utilized red, green and blue channels of original color images instead of using gray-scale image as input to avoid image detail loss. Besides, the full connection part of our network is designed in two-layers with dropout.

In the training phase, we use Euclidean Loss function in CNN training as shown in Equation 1:

$$L = \frac{1}{2N} \sum_{i=1}^{N} ||p_i^2 - t_i^2||^2 \tag{1}$$

where N is the total number of training samples, p is the predicted recognition performance score of a sample, and t is the true recognition performance score of this sample. The training target of CNN is to minimize the total loss L. The training method we use is Nesterov’s Accelerated Gradient solver [13] in CAFFE framework.

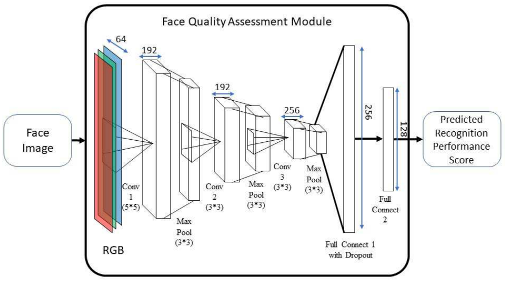

Figure 4: CNN based FQA module

3.3. Face Recognition Back-end

In order to evaluate our key-frame extraction engine, we need to setup a face recognition back-end in order to form a complete face in video recognition system. The FR back-end we use in this paper is based on GoogleNet structure and trained with identities from ChokePoint [20] and VGG dataset [14]. To imitate the real-world scenario, our DNN-based FR back-end is trained with 1,000 identities consisted of 25 known identities from ChokePoint and 975 identities from VGG face dataset. During the training phase, we picked 450 images of each identity to form the training data and the deep learning framework CAFFE with CUDNN v4 is applied to train the FR back-end. Two Nvidia Tesla K40 GPU are used to accelerate the training process. In Section 4.5 we also applied a feature-based FR back-end for comparison purpose.

4. Experiments

In this section, we first briefly introduce the video datasets used in the experiment. Then, we evaluate the performance, data volume reduction and acceleration speed of our CNN based key-frame extraction approach. The performance results of our KFE engine are also compared with existing quality based frame selection methods [15, 2, 19]. Lastly, we validate the adaptiveness of our KFE engine to different face recognition (FR) back-ends. The experiments are designed by following NIST standards [4] employing receiver operating characteristic (ROC) curve, rank dependent accuracy, and the cumulative match curve (CMC).

4.1. Dataset and experiment settings

The dataset we use is ChokePoint dataset [20]. This dataset is designed for people identification/verification under real-world surveillance conditions. Faces in this dataset all have variations in terms of illumination, pose, sharpness, as well as alignment angles. In this dataset, all videos are clipped into images frame by frame. Sixteen footages in Choke-Point dataset are one-person videos. We evenly divided them into two sets to train our FR back-end and FQA module. The other two video footages marked as P2E-S5 and P2L-S5 are recorded with people in crowd setting. We deem them as more challenging scenarios and fit our system's design target: multiple people with movement. Hence, we select these two footages to evaluate our system. The people in P2E-S5's scene make a 90-degree turn and go through the door under surveillance camera. In video P2L-S5, all people go straight through the door under camera but in a more crowded formation.

In an effort to evaluate our CNN-based approach comprehensively, we compare our scheme against a brute-force scheme performing FR on every frame as well as four representative face image quality based key-frame selection strategies: random picking, face metrics based [15], learning to rank [2] and a reference CNN based approach [19]. The random picking strategy is to pick one frame in every other N frames for the same identify (person). The face metrics based approach is to generate a quality score based on weighted square rooted combination of four metrics: resolution, sharpness, symmetry and brightness. For learning to rank method, we utilized the pre-trained module from Chen's work [2] which based on three categories of images: high quality face, low quality face and non-face. The face quality score of learning to rank approach is generated from a function which takes the weighted combination of LBP, HOG, Gist, CNN and Garbor feature vectors as input. For the referenced CNN approach, we replicated the same CNN structure in [19] and trained this reference CNN module with the same data used for training our CNN module. The other experiment settings are: Ubuntu 14.04 LTS 64-bit, OpenCV 2.4.13, OpenBR 1.1.0, CUDA 7.0, Caffe with CUDNN v4, Nvidia K40 GPU and 64GB of RAM.

4.2. Performance Results

To evaluate the performance of our key-frame extraction engine, we divide the experiments into three parts: 1) accuracy analysis on verification rate by using ROC curve; 2) accuracy analysis on rank-based recognition rate by using CMC curve; 3) application-level benefit analysis on data volume reduction and processing speedup.

4.2.1 Performance of our CNN model

In our KFE engine, we use CNN model to predict the face recognition performance value and use this value as the criteria for extracting key-frames. To evaluate our CNN model, we performed test with 2,200 face images. For an ideal FR back-end, an input face should get a performance value of 1 if it is correctly recognized, or 0 otherwise. But for practical case, errors are inevitable. As a result, a small portion of the recognizing results will lie between 0 and 1, with most of the results gathered very close to 0 or 1.

Because we employed a CNN model to generate the predicted performance score of the face image, the first thing is to validate how good the prediction is. Because this will

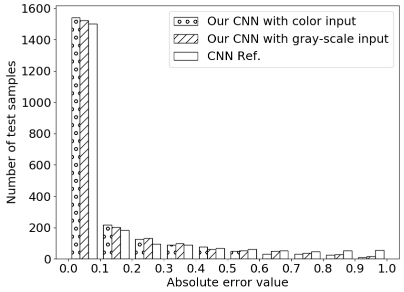

Figure 5: Absolute error histogram of real and predicted performance score across different CNN models.

affect the quality of the key frame, hence the performance of the FP back-end. In Fig. 5, we present the histogram of absolute error between actual and predicted scores across three different CNN models: 1) our CNN model with color image input; 2) our CNN model with gray image input; and 3) reference CNN model from [19]. The actual score is generated by the FP back-end, while the predicted performance score is generated by the CNN model in our KFE engine. For our CNN model with color image input, 74.13% of samples' absolute prediction error lies between [0,0.1) and 83.95% lies between [0, 0.2). Compared with the CNN model proposed in [19], in our model more test samples result in small absolute errors, which can be seen in regions [0, 0.1) and [0.1, 0,2), and less test samples result in large absolute error, which can be seen in regions [0.8, 0.9) and [0.9, 1.0). Moreover, Fig. 5 reveals that using RGB channels can get further performance gain than using gray image as input. Overall, our CNN based prediction model reaches a correlation value of 85.44%, which means the predicted performance score has a strong correlation to actual FR performance value, validating the effectiveness of our proposed FQA module.

4.2.2 Accuracy of CNN-based key-frame extraction

Next, our CNN based KFE approach is compared with performing FR with the following frame selection strategies: 1) all frames; 2) randomly picked frames; 3) face quality metrics based approach; 4) Learning to rank; 5) reference CNN based approach. The results are presented as ROC curves in Fig. 6 and Fig. 7, and CMC curves in Fig. 8 and Fig. 9,

From Fig. 6 and Fig. 7, we can see our CNN-based approach outperforms all other approaches we evaluated in term of Area Under the Curve (AUC) value. Compared with performing FR on every frame, our approach improved AUC value by 0.66% in P2E-S5 video and 5.96% in P2L-S5 video, respectively. We also compared the verification accuracy in terms of True Positive Rate (TPR) of all six frame selection methods at 2.5% False Positive Rate (FPR).

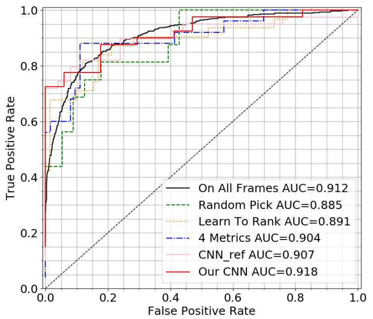

Figure 6: ROC curve of different key-frame selection engines using P2E-S5 video

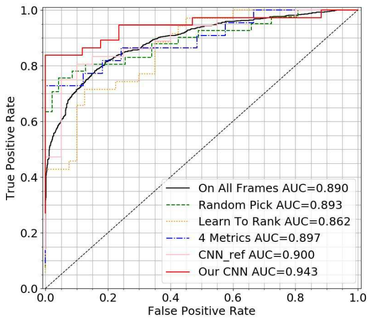

Figure 7: ROC curve of different key-frame selection engines using P2L-S5 video

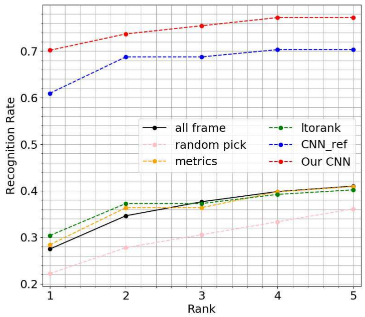

Figure 8: CMC curve of different key-frame selection engines using P2E-S5 video

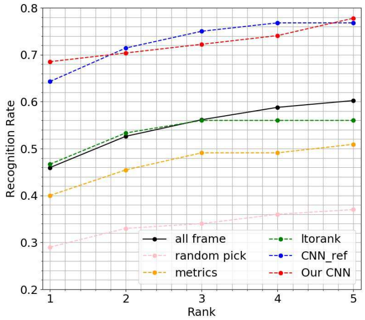

Figure 9: CMC curve of different key-frame selection engines using P2L-S5 video

Method	P2E-S5	P2L-S5	Difference
On All Frames	0.534	0.547	0.013
Random Pick	0.438	0.707	0.269
Learning to Rank [2]	0.677	0.429	0.248
Face Metrics [15]	0.600	0.727	0.127
Ref. CNN based [19]	0.718	0.472	0.246
Ours	**0.725**	**0.838**	**0.113**

Table 1: TPR value comparison across difference key frame extraction engines @ 2.5% FPR

In other words, we tested the frame selection systems under a real application scnario, to compare the success possibility for getting a true recogniton while having a small false recogniton trade-off. The reason for comparing TPR at 2.5% FPR is that the ROC curves are based on extracted key-frames, which are relatively small numbered. As a result, all ROC curves seem less continous. Hence, 2.5% is the smallest number possible to choose in Fig. 6 and Fig. 7, which fits the purpose of cross comparing all 6 methods. From Table 1, we can conclude that our CNN based approach achieved highest accuracy on both video footages at 2.5% FPR. Besides, the performance difference on these two videos is also smaller than other four key frame selection methods, only worse that all-frame method. This indicates that our CNN based approach has a better adaptiveness to different video conditions.

We also compared the cumulative recognition accuracy of all methods presented as the Cumulative Match Curve (CMC). Different from ROC curve, CMC curve focuses on the performance of extracting identity of wanted people based on possibility value ranked output of FR system. In real world application, CMC reflects the success rate of getting right identification on FR system's Top-N output. In Fig. 8 and Fig. 9, compared with FR on all frames, both CNN based approaches stand out of all methods and bring highest recognition rate improvement from Top-1 to Top-5 ranks. The random picking method performs worst since it does not utilize any face image quality information. Learning to rank and metrics based approaches stand in the middle and reached accuracy comparable to that of every-frame based FR. To conclude, we found that our CNN based frame selection approach achieved best and most stable effects in improving the accuracy of face recognition back-end.

The reason for our CNN based approach achieved best performance is: first, the all frame based and random picking approaches do not consider face quality at all, so these two methods get poor performance; second, the metrics based approach considers face quality but follows a simple, primitive weighted summation on global resolution, brightness, etc; third, the learning-to-rank approach considers fine-grained features such as LBP, HOG, Garbor, etc. Hence, it showed a better performance stability compared with metrics based approach in our experiments. The CNN based approaches achieved outstanding performance since the CNN does not need predefined features but "decides" which kind of features, or which convolutional kernels, are important during the training stage. Lastly, since we improved the CNN structure, our approach achieved best performance in all methods expect in Fig. 9, rank 2-4 accuracy for P2L-S5 video, which is slightly worse than the referenced CNN method. The reason is, our FQA module's CNN training is still not 100% perfect. Hence, for some face images, we could have mis-predictions which make non-best faces being picked and down-grade the overall FR performance.

4.3. Data volume reduction and processing speed

In real application scenario, the reduction on data volume helps saving the network bandwidth and storage space. On the other hand, processing speed is another concern since the existence of real-time processing demand, e.g., to catch up with the frame rate of 24 to 30FPS of surveillance camera. As shown in Table 2, our CNN based key frame extraction (KFE) engine is able to reduce the original data volumes for the two video sequences by 93.65% and 92.85%, respectively. This means we can reduce more than 90% workload of performing FR with complex DNN such as the GoogleNet we use in back-end. Among all four quality based frame selection methods, our CNN based approach achieved the highest data reduction ratio. And our CNN based KFE engine also reached higher than real-time processing speed at 55.31 FPS and 51.43 FPS on P2ES5 and P2LS5 footages, respectively.

Video	Total Frames	Method	Extracted Key-Frames	Reduction Ratio
P2E-S5	898	Learning to Rank	102	88.64%
		Face Metrics	88	90.20%
		Ref. CNN	64	92.87%
		ours	57	93.65%
P2L-S5	755	Learning to Rank	75	90.07%
		Face Metrics	55	92.71%
		Ref. CNN	56	92.77%
		ours	54	92.85%

Table 2: Data volume reduction comparison

Finally, we selected one identity from the video sequence to demonstrate the sample output of our key-frame extraction engine, which is shown in Fig. 10. The person was first detected in the 527th frame and continued to be detected until the 538th frame. Among the faces detected from Frames 527 to 538, our KFE engine picked face in Frame 537 as the best quality face, which is marked with red box in the figure. Hence, the corresponding frame is selected as the key frame for that identity.

4.4. Timing analysis

We also conducted timing analysis between our KFE based and all-frame based FR approaches to see if KFE

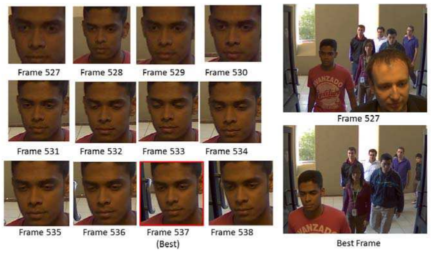

Figure 10: Sample Output of our Key-Frame Extraction Engine

can benefit the whole processing chain of FR, from font-end's key-frame analysis and back-end's FR. For key-frame analysis, it typically takes 0.019s for a face image to pass through KFE CNN and get face quality result. And it takes 0.146s to perform FR on a face image by passing it through the GoogLenet based CNN. As we can see in Tables 3 and 4, although KFE based approach needs additional key-frame analysis time before performing FR, the back-end only needs to perform FR on faces in extracted key-frames. Overall, in videos P2E-S5 and P2L-S5, with limited overhead time of 11.742s and 11.096s, we achieved about 4.5 times speedup in terms of the overall processing time. Hence, our KFE approach can benefit the FR processing with relatively small overhead caused by performing key-frame analysis.

Processing Stage	Processing Time of Each face	Number of Faces need to be Processed		Accumulative Time Difference
		KFE Approach	All-Frame Approach	
Key-Frame Analysis at Front-end	0.019 s	618	Not Needed	+11.742 s
Face Recognition at Back-end	0.146 s	57	618	-81.906 s
Total Process Time		20.064 s	90.228 s	-70.164 s

Table 3: Timing analysis between KFE and all-frame based approaches on P2E-S5 video

Processing Stage	Processing Time of Each face	Number of Faces need to be Processed		Accumulative Time Difference
		KFE Approach	All-Frame Approach	
Key-Frame Analysis at Front-end	0.019 s	584	Not Needed	+11.096 s
Face Recognition at Back-end	0.146 s	54	584	-77.380 s
Total Process Time		18.980 s	85.264 s	-66.284 s

Table 4: Timing analysis between KFE and all-frame based approaches on P2E-S5 video

4.5. Adaptiveness to different FR back-end

To validate our approach is adaptive to different FR back-ends, especially for FR back-end of less accuracy, we designed a separate set of experiments. Since feature based approach is another representative FR methodology, we applied 4SF algorithm using LBP and SIFT face features with OpenBR framework [9]. The number of training samples for each identity is set to 50, targeting the scenario when there are only limited number of training images for each identity. We reduced the number of training samples from 450 to 50 for each identity in our CNN based FR back-end as well for fair comparison. The dataset and other settings are the same as those described in Section 4.1.

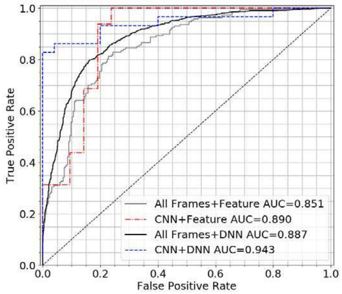

Figure 11: ROC curve of using our CNN based frame selection engine with feature based back-end under P2E-S5 video

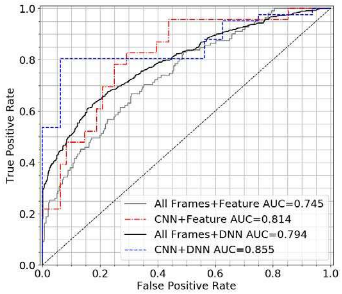

Figure 12: ROC curve of using our CNN based frame selection engine with feature based back-end under P2L-S5 video

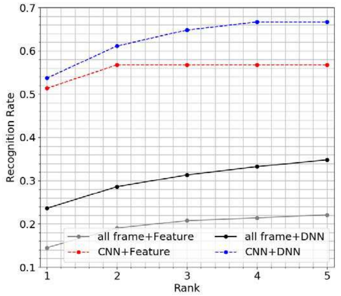

Figure 13: CMC of using our CNN based frame selection engine with feature based back-end under P2E-S5 video

As seen in Fig. 11 and Fig. 12, generally our CNN based KFE engine achieved better results compared with performing FR on every frame in both footages. Also in Figs 13 and 14, both our CNN based approach brought recognition rate improvement on top-1 and top-5 ranks. To conclude, we demonstrate that our approach is also effective on feature based FR back-end, which means our CNN based KFE is adaptive to different FR back-ends.

5. Conclusions

Face in video recognition is an important method in extracting identity under video surveillance or large volume video footage processing cases. In this paper, we presented a CNN based key-frame extraction (KFE) engine

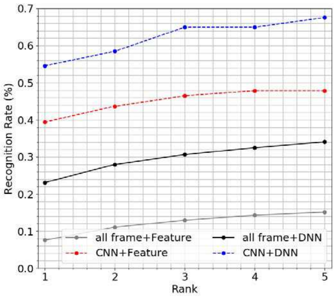

Figure 14: CMC of using our CNN based frame selection engine with feature based back-end under P2L-S5 video

to improve the accuracy and processing speed of face in video recognition. The experiments show our KFE engine achieved the best performance among comparable algorithms, higher than real-time processing speed and over 90% data volume reduction rate. With the ROC and CMC analysis, we demonstrate that correlating frame selection with the information from the face recognition back-end is a better, more adaptive approach than only evaluating the absolute quality of face image itself. What's more, from the timing analysis we can conclude that performing key-frame selection is effective in reducing the overall processing time of face recognition. We employed graphic processing unit (GPU) throughout our framework, from key frame front-end to face recognition back-end. GPU acceleration is essential for our framework to be able to achieve real-time (or even faster than real-time) processing speed of the video streams, even at HD resolution. Besides, we want to note that our KFE engine is functional on CPU-only platforms as well. For future works, we will continue to improve the structure of our CNN model to achieve higher face quality assessment performance on accuracy and speed, as well as employing more advanced neural network models.

Acknowledgement

This material is based upon work supported by the Center for Identification Technology Research and the National Science Foundation (NSF) under Grants No.1068055 and 1650503. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] K. Anantharajah, S. Denman, D. Tjondronegoro, S. Sridharan, C. Fookes, and X. Guo. Quality based frame selection for face clustering in news video. In *Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[2] J. Chen, Y. Deng, G. Bai, and G. Su. Face image quality assessment based on learning to rank. *IEEE Signal Processing Letters*, 22(1):90–94, 2015.

[3] T. I. Dhamecha, G. Goswami, R. Singh, and M. Vatsa. On frame selection for video face recognition. In *Advances in Face Detection and Facial Image Analysis*, pages 279–297. Springer, 2016.

[4] P. Grother and M. Ngan. Face recognition vendor test (frvt): Performance of face identification algorithms. *NIST Interagency report*, 8009(5), 2014.

[5] A. Hadid and M. Pietikainen. Selecting models from videos for appearance-based face recognition. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 304–308. IEEE, 2004.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[7] R. R. Jillela and A. Ross. Adaptive frame selection for improved face recognition in low-resolution videos. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 1439–1445. IEEE, 2009.

[8] H.-I. Kim, S. H. Lee, and Y. M. Ro. Face image assessment learned with objective and relative face image qualities for improved face recognition. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4027–4031. IEEE, 2015.

[9] J. C. Klontz, B. F. Klare, S. Klum, A. K. Jain, and M. J. Burge. Open source biometric recognition. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8. IEEE, 2013.

[10] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–I. IEEE, 2002.

[11] W. Liu, Z. Li, and X. Tang. Spatio-temporal embedding for statistical face recognition from video. *Computer Vision–ECCV 2006*, pages 374–388, 2006.

[12] K. Nasrollahi and T. B. Moeslund. Face quality assessment system in video sequences. In *Biometrics and Identity Management*, pages 10–18. Springer, 2008.

[13] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[14] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[15] X. Qi and C. Liu. Gpu-accelerated key frame analysis for face detection in video. In *Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference on*, pages 600–605. IEEE, 2015.

[16] U. Saeed and J.-L. Dugelay. Temporally consistent key frame selection from video for face recognition. In *Signal Processing Conference, 2010 18th European*, pages 1311–1315. IEEE, 2010.

[17] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

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

[19] S. Vignesh, K. M. Priya, and S. S. Channappayya. Face image quality assessment for face selection in surveillance video using convolutional neural networks. In *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*, pages 577–581. IEEE, 2015.

[20] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B. C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 81–88. IEEE, June 2011.

[21] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.

[22] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 866–870. IEEE, 1998.