Compression Artifact Mitigation for Face in Video Recognition

Xuan Qi[a] and Chen Liu[b]

[a,b]Clarkson University, 8 Clarkson Ave., Potsdam, NY, US

ABSTRACT

Face in video recognition (FiVR) is widely used in video surveillance and video analytic. Various solutions have been proposed to improve the performance of face detection, frame selection and face recognition in FiVR systems. However, all these methods have a common inherent "ceiling", which is defined by the source video's quality. One key factor causing face image quality loss is video compression standards. To address this challenge, in this paper, first, we analysis and quantify the effects on the FiVR performance due to video compression; secondly, we propose to use deep learning based model to mitigate artifacts in compressed input video. We apply the image based convolutional auto-encoder (CAE) to extract the features of input face images and restore them towards less artifacts. From the experimental results, our approach can mitigate artifacts on face images extracted from compressed videos and improve the overall face recognition (FR) performance by as much as 50% in TPR (True Positive Rate) at the same FPR (False Positive Rate) value.

Keywords: FiVR, face recognition, artifacts mitigation, video compression, deep learning, CAE

1. INTRODUCTION

Video compression techniques are widely applied to reduce redundant data in video, make efficient video storage and live video streaming through network without consuming significant bandwidth,but still keep a reasonable video quality. Many video compression standards like MPEG-4[1] and H.264[2] use spatial compression and temporal motion prediction in their algorithm implementation. Even though being able to reduce video size to fit limited network bandwidth and the capacity of data storage devices, video compression with non-lossless algorithm will cause trade-offs like video quality reduction, or compression artifacts. The quality loss would be more obvious if we put a cap on the data-rate or bandwidth, which would result in low visual quality.

Video compression can cause both spatial and temporal artifacts. The most common compression artifacts are illustrated and categorized in Fig.1. For spatial artifacts, the video compression can cause blocky, blurry and speckling effects on objects in videos. As for temporal artifacts, the objects in video can be floating or flicking due to the compression. Sometimes multiple or hybrid artifacts can occur at the same time. For example, on the right side of Fig.1, the top face image has blocky artifact, which is spatial; the middle image is blurred, which is spatial; and the bottom one has blurry and floating, which relate to both spatial and temporal artifacts.

Face in video recognition (FiVR) is widely used in video surveillance and video analytic. Various solutions have been proposed to improve the performance for different stages of FiVR system, from face detection,[3,4] to frame selection[5,6] and face recognition (FR) engine.[7,8] The work on face detection targets detecting faces of non-ideal quality and various poses, providing more samples to FR engine. The work on frame selection targets selecting face images with better quality, such as higher resolution or better pose. As a result, the poor quality faces can be excluded from down-grading the FR performance. The work on FR engine targets improving the robustness of FR models and making them capable of handling non-ideal or even low-quality faces while still output correct and high confidence FR results. But all these methods have a common inherent "ceiling", which is defined by the source video's quality. The compressed video is normally the input to the FiVR system. We can intuitively speculate the compression artifacts will affect the follow-up processing such as the detected face in

Further author information: (Send correspondence to Xuan Qi)
Xuan Qi: qix@clarkson.edu
Chen Liu: cliu@clarkson.edu

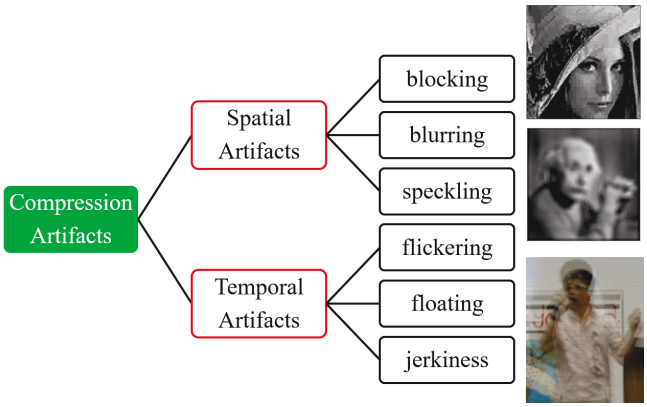


Figure 1: Video Compression Artifacts

video sequence or downgrade the performance of FR due to quality loss when compared with processing directly on the uncompressed videos. And the situation could be even worse if we perform FR on the videos with higher compression ratio.

In this work, we propose to use deep learning based model to mitigate artifacts in compressed input videos, targeting towards effectively improving the FR performance of FiVR system. This work is novel in applying the image based convolutional auto-encoder (CAE) to extract the features of input face images and restore them towards less artifacts. The main contribution of this work is to use the CAE with skip connection as the core of artifact mitigation engine and integrate it into the FiVR framework.

The rest of the paper is organized as follows: In Section 2, we review related works including image and video based super-resolution and image-based artifact mitigation. In Section 3, we describe our video-based, face image targeted artifact mitigation approach. In Section 4, we present experimental results and data analysis of our approach. We conclude in Section 5.

2. RELATED WORKS

In this section, we first review existing researches that relate to artifact mitigation. Next, we briefly review related research on machine learning models that can restore images with noise or artifacts. Then, we discuss the difference between our work with the existing works.

2.1 Existing video/image artifact mitigation approaches

The first area relates to our work is super-resolution and artifacts reduction for single frame or static image. Dong et. al. proposed a Deep Neural Network (DNN) based image super-resolution.[9] In this work, the low-resolution image is first passed through convolutional layer to extract feature maps. Then, a non-linear mapping stage is performed to match the features extracted from high-resolution image as close as possible. Finally, the last layer combines re-mapped feature maps to construct the high-resolution image. Kim et. al. proposed a recursive DNN-based approach.[10] They applied a 3-stage structure, which is feature extraction, feature mapping and reconstruction. But the difference is that they used a recursive structure in feature mapping stage, and achieved better performance than that of Dong et. al.[9] One step further, a couple of other research works[11,12] deal with multiple artifacts like blocky and blurry caused by image compression. In these works, deep convolutional neural network (CNN) is applied and a feature enhancement stage is added into the feature re-mapping stage.

The second area which more directly relates to our work is video-based super-resolution. Since videos contain both spatial and temporal information, which is relatively more complex than images, Kappeler et. al. proposed a CNN-based video super-resolution approach which has spatial fusion architecture.[13] In their work, they presented three different fusion strategies: fusion at feature extraction stage, fusion at feature re-mapping stage and fusion at final feature combining stage. Caballero et. al. also proposed a video super-resolution approach[14] using a different spatial fusion method, 3-D convolution. The 3-D convolution forces weights of different feature maps

extracted from different frames to be shared temporally. By using 3-D convolution, the spatial and temporal information can be better shared across different frames, and the computation overhead can also be reduced since the result from previous frames can be reused. Overall, both methods achieved better performance than single frame based super-resolution.

2.2 Machine learning models related to image restoration

For machine learning based image processing, researchers normally apply CNN to achieve satisfying performance across a variety of tasks such as object classification, face recognition,[15,16] object detection,[17,18] and image segmentation.[19,20] By applying the philosophy of auto-encoder, researchers came up with convolutional auto-encoder (CAE) to overcome the challenges like image de-noising, image super-resolution and video super-resolution. Different from traditional CNN structure which only performs feature extraction, the CAE has additional de-convolutional structure which performs image reconstruction based on extracted features. Hence, the convolutional component in CAE acts as encoder that transforms the input image into the latent space. Then, de-convolutional component acts as decoder which outputs a restored image based on extracted latent vectors. Besides, Pu et.al.[21] applied an image-based variational auto-encoder (VAE), a variation of CAE, and evaluated their approach with different image-based tasks including classification, labeling and caption creation. Different from the CAE which only focuses on reconstruction loss between input and output images, the VAE model has a second focus on the distribution of latent space. In VAE, the input image is first encoded as distribution over a sample point in the latent space; second, a point is sampled from the latent distribution; then, the reconstruction will be performed based on this sampled point.

2.3 Relationship with our work

From the brief review above, we can see that there are many existing works proposed on whole image/video-frame based solutions, targeting image quality and improving the Signal to Noise Ratio (SNR). But our application field is specifically towards FiVR. Technical wise, our work is video-based artifact mitigation, which is different from image-based super-resolution, image-based artifacts mitigation, or video-based super-resolution. Moreover, to improve the FiVR performance, we focus on restoring face images with artifacts detected from compressed videos. Hence, our approach utilizes face features and restores face images based on extracted features, but not deals with other information included in the whole video frame such as background or object, which are not related with our face recognition task.

3. OUR APPROACH

In Fig.2, we present a top-level block diagram of our approach. First of all, we perform face detection from the input compressed videos which contain artifacts already. Then, we perform the artifact mitigation on the detected face by using our DNN-based artifact reduction engine. Finally, the face images with reduced artifacts are forwarded to further processing including face quality assessment and face recognition. One important detail is that we perform artifact mitigation after face detection, but not performing artifact reduction on the whole video frame. The reason for this is two-folds. First, we can reduce the computation overhead since we don't need to compute over the area of an entire frame, and we don't need to forward a large size image into the deep neural network. Second, the artifact mitigation engine would be more "focused" when we use detected face images as input, as video frame with background information is unrelated to our FR task. As a result, the DNN model in artifact mitigation engine would only extract useful features of faces, but not the features from other areas of the video frame, such as non-face objects or background features. Then, the following frame selection and FR engine will perform the operation on the restored face images, which is the output of the proposed artifact mitigation engine.

The detailed framework of our face image artifact mitigation engine is presented in Fig.3. The first stage is face detection, which crops faces from original video frames. Then, the detected face images will be re-scaled to the input size of the neural network. In this work, we set the input size to be 224×224. One reason for this size setting is that the target of artifact mitigation is to improve the FR performance. Hence, we directly target the FR DNN and re-scale them to the same input size of FR DNN, so that they can connect seamlessly. Even though our current backend is VGGNet, our framework can work for other FR DNN models since the input size

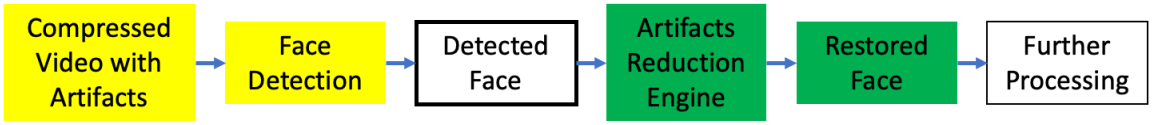


Figure 2: Block diagram of our approach

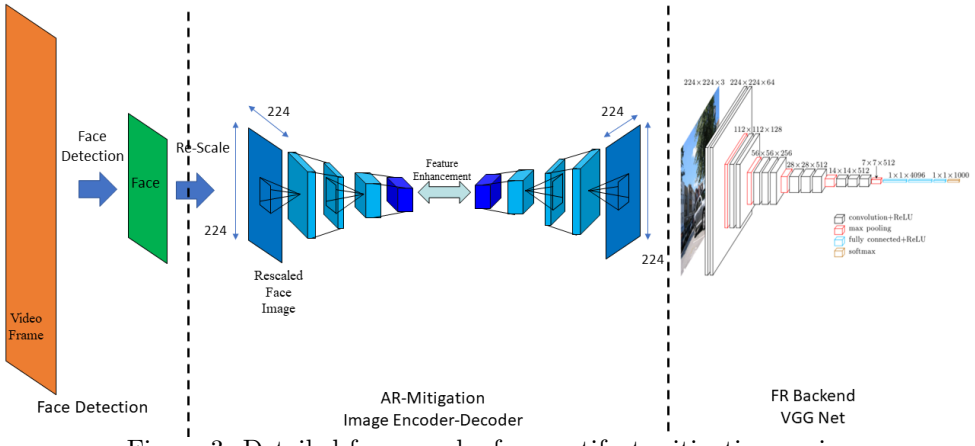


Figure 3: Detailed framework of our artifact mitigation engine

of GoogLenet V1 and ResNet-101/152/50 is also 224×224. Please note we also have the option to set the size to 299×299 to adapt to GoogLenet V3-5 versions. In the middle part of our engine, we use an image-based auto-encoder to extract features of input face image and reconstruct the face images with less artifacts based on the enhanced features. The left side uses convolutional operations to generate smaller but more feature maps, while the right side applies de-convolutional operations which rebuild the larger face images based on extracted feature maps. Lastly, the FR would be performed on restored face images.

The core of our artifact mitigation framework is the DNN with convolutional auto-encoder structure, which is shown in more detail in Fig.4. The stages shown in orange is the convolution operation which performs feature extraction on input face image. The stages in green is the de-convolution operation, which rebuilds the face image based on extracted feature maps. The reason for using convolutional auto-encoder is, the convolution and de-convolution structure is capable of utilizing useful face features and stops unrelated features or noise caused by compression artifacts being passed on to the image reconstruction. In other word, if the model is trained with more iterations, the convolution extracts useful features relate to face structure, shape and details. And the unrelated features which caused by artifacts would be filtered out. Then, the de-convolution stage can rebuild the face image based on extracted useful face features. We can also see that the convolution and de-convolution parts are symmetric in the number of operation stages and feature map sizes at different level. Hence, we further add the skip connection, which combines the input images or feature maps with the reconstructed images or feature maps. The benefit of introducing skip connections is that we can keep the useful information in original input and forward it to reconstruction stage to achieve better performance.

Another detail in model training is that we use the reconstruction loss, which is the mean squared error (MSE) as the target loss function. As shown in Equation 1, the MSE loss between reconstructed face image $\hat{I}(m,n)$ and input face image $I(m,n)$ is computed as the mean squared difference between each pixel at the same position (m,n):

$$Loss_{MSE} = \frac{1}{MN} \sum_{n=1}^{M} \sum_{m=1}^{N} [\hat{I}(m,n) - I(m,n)]^2 \tag{1}$$

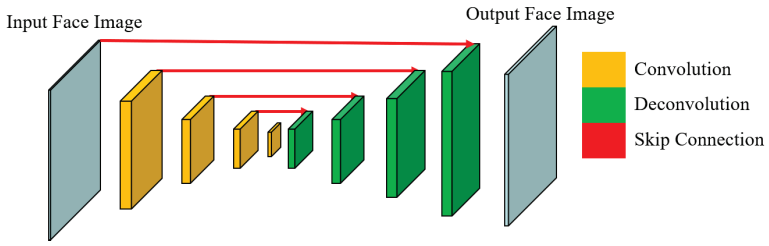


Figure 4: Convolutional auto-encoder with skip connection

4. EXPERIMENTS

In designing the experiments, we want to quantify the FR performance loss caused by artifacts first to show the FR performance impact due to video compression. Then, we restore the face images from compressed video with artifacts, and compare the overall performance between FR on restored face images and on un-restored face images.

For the evaluation of face in video dataset, we used 1401 videos from stabilized camera set in PaSC dataset,[22] and compressed them with H.264 and MPEG4 codec standards. For the training and testing of our artifacts mitigation engine, we varied different combinations of codec and bit-rate. Then, we examined the FR performance with ROC and DET curves under different settings. The ROC curve emphasize on the accuracy in terms of TPR (True Positive Rate) under specific tolerance of error rate in terms of FPR (False Positive Rate). The DET curve emphasizes on the trade-offs of FPR (False Positive Rate) over FNR (False Negative Rate). For FR engine, we use VGG Face Descriptor in training and testing. Other details include we use PyTorch 1.1 as the machine learning framework, Nvidia Titan XP GPU, 64GB system RAM, two Intel Xeon-2650v3 10-core CPUs and Ubuntu 18.04 64bit as experiment platform.

4.1 Impact of Video Compression

To clearly show the impact on FR performance caused by video compression, we compressed 2802 test videos from PaSC dataset using H.264 and MPEG4 standards with different compression bit-rates. In ROC and DET curves below, we can see that video compression causes notable FR performance loss, especially under low bit rates. For example, in the ROC curve of Fig.5, there is an 18% performance loss for TPR at FPR of 0.2 between uncompressed and H.264 512kpbs compression, which is observable and cannot be ignored. The AUC is decreased as well. For example, the AUC of H.264 with 512kbps is 8% lower compared with that of uncompressed case. In DET curve, we can see an obvious FNR and FPR increase due to compression. The curve of FR on uncompressed video, which is the solid curve, has a lower than 0.5 FNR and lower than 0.6 FPR. But if the videos are compressed with H.264 at 512kbps, the FNR increases by 20% and the FPR increases even higher. On a side note, we can observe that at the same bit-rate, the AUC of H.264 is higher than that of MPEG4, and the DET curve of H.264 is also lower, indicating H.264 compression standard causes less performance loss than using MPEG-4 standard. To sum up, it is evident that compression has a significant impact on the performance of FiVR.

4.2 Result of Artifacts Mitigation

First, we compared the performance across different auto-encoder structures for our artifact mitigation engine in Fig.6. For the model training we used compressed videos with H.264 standard at 2048kbps bit-rate and applied L1 loss as the performance metric. From this figure we can see that the convolutional auto-encoder (CAE) with skip connection has the lowest L1 absolute loss, compared with other three representative auto-encoder structures. Specifically, CAE L1 loss is around 50% lower than that of adversarial auto-encoder, and around 30% lower than that of convolutional auto-encoder without skip connection. This is because the skip connection passes information from feature extraction layers and makes de-convolution layers being able to combine original features with reconstructed features and achieve lower loss.

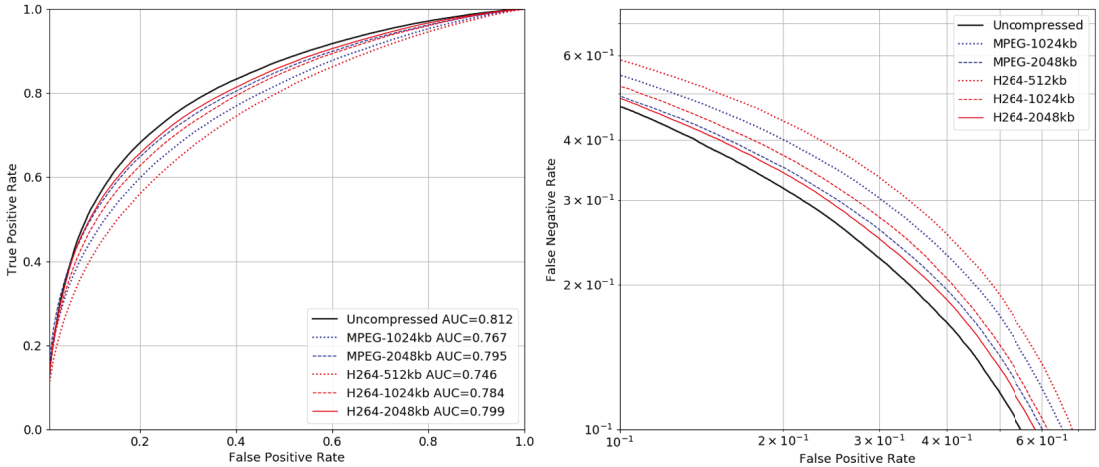


Figure 5: Comparison on FR performance loss due to compression (ROC curves on the left and DET curves on the right)

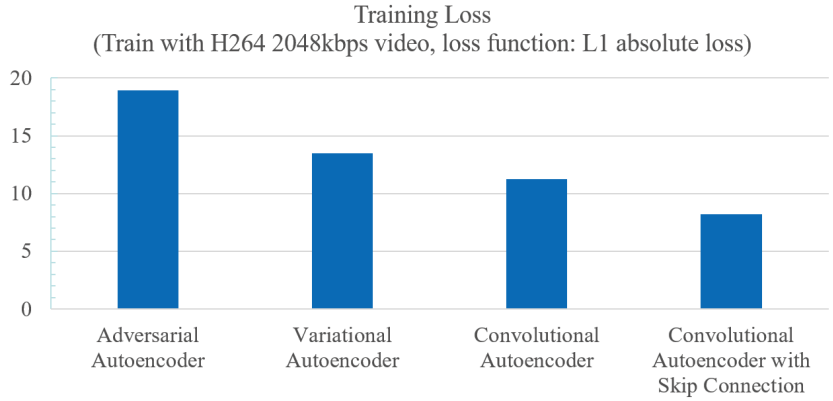


Figure 6: Comparison of L1 loss of different model structures

Next, we present a cross comparison study using two video compression standards (H.264 and MPEG4) at two different bit-rate (2048kbps and 1024kpbs) to examine different training and testing combinations, the results of which can be found in Fig.7. First of all, we can observe four working combinations labeled as green, i.e., if we train our model with H.264 compressed videos and test on MPEG4 compressed videos, we can achieve positive mitigation effect. Actually, under the same bit-rate, H.264 standard has been shown to be able to keep higher video quality and being a more efficient standard than MPEG4.[23] Hence, if we train our mitigation engine with better compression standard, the model can learn more face features and being able to restore faces in videos compressed with less efficient compression standard. This even applies to the case when we train the model with H.264 at lower data-rate, we can still improve the performance of MPEG4 video at higher data-rate.

Secondly, there are six red blocks on the upright of Fig.7 where we have negative mitigation effect. Actually, training on videos compressed with less efficient standard such as MPEG4 and test on videos compressed with better standard like H.264 will not achieve the artifact mitigation effect we desire, across different bit rates. The same scenario happens on using the model trained on videos with lower bit rate and applying towards videos with higher bit rate in hope of achieve mitigation effect, with the same compression standard. Hence, these configurations mean the model can't learn enough features with poorer video. Lastly, if we train the mitigation engine using the same compression standard at the same bit rate as the testing video, we do not achieve obvious FR performance improvement across the board. We speculate those videos for training and testing have similar level of details, as a result, the model couldn't learning more details to restore faces from compressed videos.

In Fig.8, we present the results of FR performance improvement by applying compression artifact mitigation.

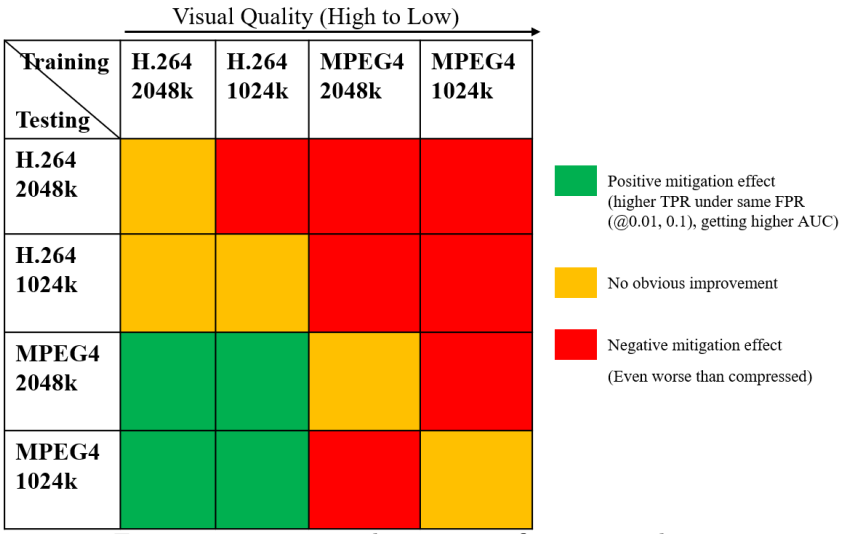
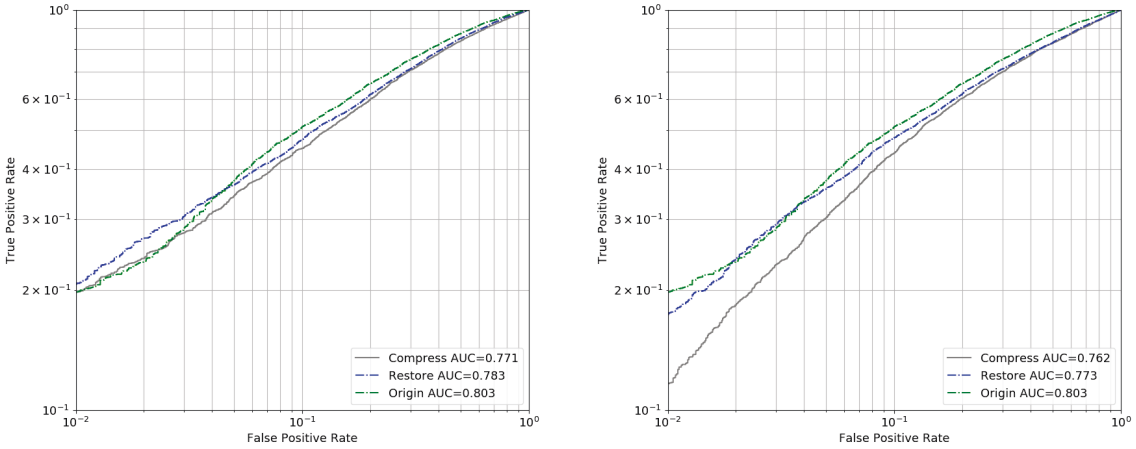


Figure 7: Training and testing configuration selection

Based on the study from Fig.7, the mitigation model is trained with higher quality video of H.264 at 2048kbps, and tested with MPEG videos at 2048kbps and 1024kbps, separately. From the ROC curve we can see that the video compression downgrades the FR performance. From Fig.8a, the compression at 2048kbps causes a 4% loss in AUC compare with that of FR on original video. And in Fig.8b, this loss in AUC is further increased to 5% at a lower bit rate. And a more obvious loss can be seen in TPR value at 1% FPR, where the 2048kpbs case has a TPR of 0.2 and 1024kbps case falls much below 0.2 to close to 0.1, which means a nearly 50% performance loss. But with our artifacts mitigation engine, the FR performance is restored to very close to the performance from uncompressed video, especially in low FPR area. For example, at the same 1% FPR location, both ROC curves in blue has a close to 20% TPR, which means our mitigation engine largely restores the FR performance. Furthermore, to show the performance improvement more clearly, we compared the absolute TPR values in Fig.9 for compressed, restored and original videos under 0.01 and 0.1 FPR, respectively. We can clearly see that the artifact mitigation engine improves the video quality and makes the FR performance higher than performing FR on original compressed video, and even getting close to FR on original uncompressed video. In Fig.10, we present the result of performance comparison in DET curves. We can see that the video compression can bring more trade-offs in FPR and FNR, especially at lower bit rate. But with our artifacts mitigation engine, the FPR and FNR are both reduced.

What's more, we tested our artifact mitigation engine with MPEG4 compressed videos at an even lower bit-rate of 512kbps. From the Fig.11 below, we can clearly see that the compression artifacts mitigation engine works on even worse quality videos, as well as on MPEG4-2048kbps and MPEG4-1024kps compressed videos. Even though the AUC metric of our approach downgrades as the video quality deteriorates, the dropping trend of AUC value for performing FR on restored videos is less sharp than that of performing FR on compressed videos, which indicates that our approach is capable of handling low quality videos with even higher compression ratio.

Lastly, we present the visual effect of compression artifacts mitigation, where the face samples are also from PaSC dataset. In Fig.12, we can see the face images in compressed video are very blocky and vague. But with our artifacts mitigation, the face images in restored video are much less blocky and closer in quality to faces in original uncompressed video. What's more, we use a face quality assessment (FQA) engine from our previous research[6] to quantify the quality of the face samples. The score is normalized to 0.0 to 1.0, from low to high. For the most left face in red box, the face from compressed video has a quality score of 0.52, and the restored one has a quality score of 0.69, which is much closer to the original face's quality score of 0.76.



(a) MPEG 2048k video (b) MPEG 1024k video

Figure 8: ROC curve of test on compressed video

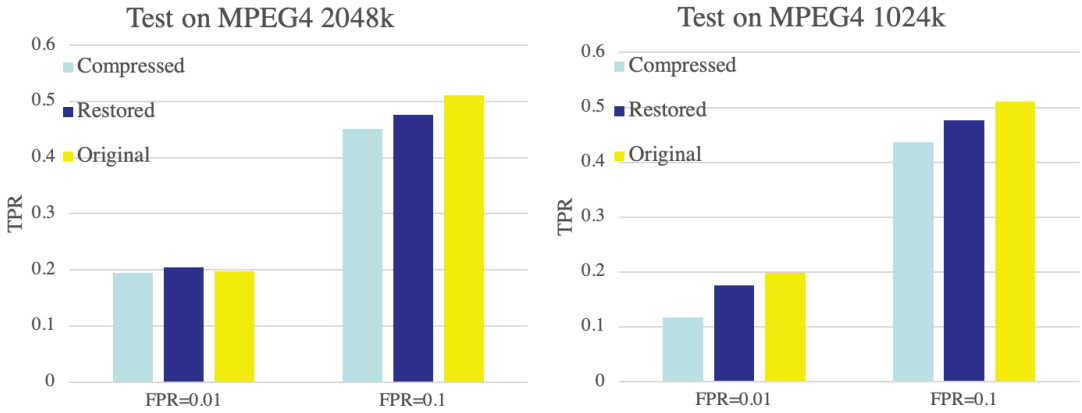


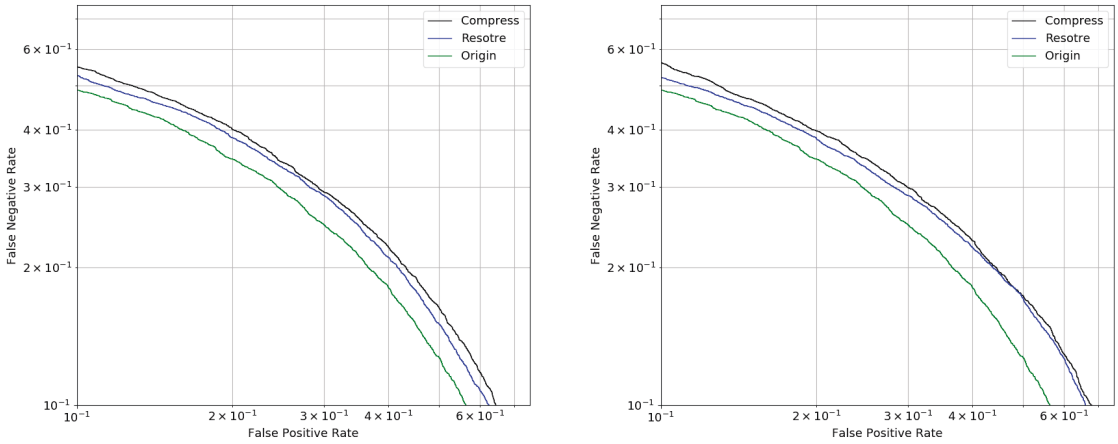
Figure 9: Comparison of absolute performance value

5. CONCLUSION

In this work, we introduce an artifact mitigation engine targeting face in video recognition (FiVR) system. First, from the quantitative study on the effect of video compression on the performance of face recognition (FR), we can see that the compression can make the overall FR performance lower comparing with performing FR on uncompressed videos. Secondly, we propose a deep learning model which applies convolutional auto-encoder (CAE) with skip connections to restore face images detected from compressed videos. From the experimental results we can see that our approach can restore the quality of face from highly compressed video streams with low visual quality, hence effectively mitigate the compression artifacts on faces and improve the overall FR performance. For future works, we plan on continuing to improve the model to utilize temporal information cross multiple video frames.

ACKNOWLEDGMENTS

This material is based upon work supported by the Center for Identification Technology Research and the National Science Foundation under Grant No. 1650503, as well as the National Science Foundation under Grant No. CNS-1626360. The Titan XP GPU used for this research was donated by the NVIDIA Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and



(a) MPEG 2048k video (b) MPEG 1024k video

Figure 10: DET curve of test on compressed video

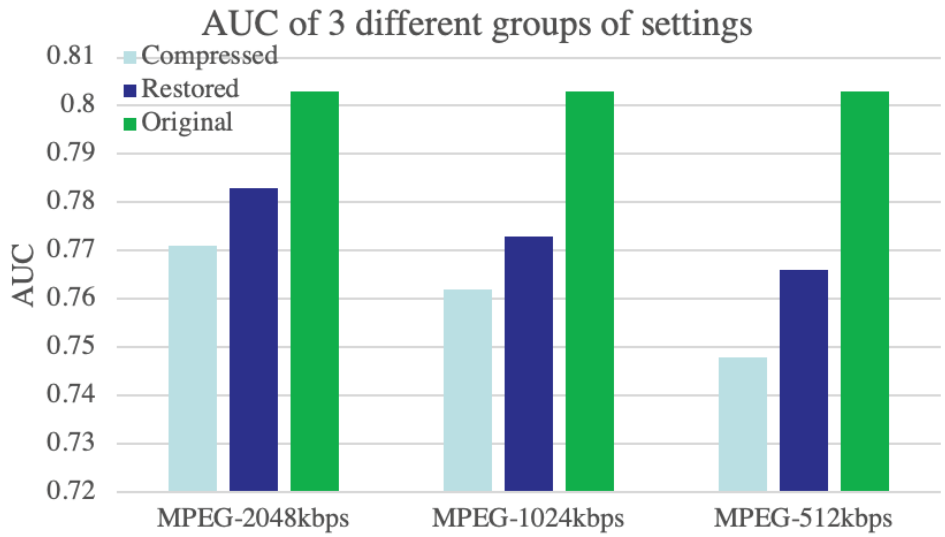


Figure 11: Comparison of AUC values across three different MPEG4 bit-rates

Figure 12: Visual effects of compression artifacts mitigation

do not necessarily reflect the views of the Center for Identification Technology Research, the National Science Foundation or the NVIDIA Corporation.

REFERENCES

[1] Le Gall, D., "Mpeg: A video compression standard for multimedia applications," *Communications of the ACM* **34**(4), 46–58 (1991).

[2] Richardson, I. E., [*The H. 264 advanced video compression standard*], John Wiley & Sons (2011).

[3] Jiang, H. and Learned-Miller, E., "Face detection with the faster r-cnn," in [*2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*], 650–657, IEEE (2017).

[4] Ranjan, R., Patel, V. M., and Chellappa, R., "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(1), 121–135 (2017).

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

[6] Qi, X., Liu, C., and Schuckers, S., "Boosting face in video recognition via cnn based key frame extraction," in [*2018 International Conference on Biometrics (ICB)*], 132–139, IEEE (2018).

[7] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L., "Sphereface: Deep hypersphere embedding for face recognition," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 212–220 (2017).

[8] Wen, Y., Zhang, K., Li, Z., and Qiao, Y., "A discriminative feature learning approach for deep face recognition," in [*European conference on computer vision*], 499–515, Springer (2016).

[9] Dong, C., Loy, C. C., He, K., and Tang, X., "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence* **38**(2), 295–307 (2015).

[10] Kim, J., Kwon Lee, J., and Mu Lee, K., "Deeply-recursive convolutional network for image super-resolution," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 1637–1645 (2016).

[11] Dong, C., Deng, Y., Change Loy, C., and Tang, X., "Compression artifacts reduction by a deep convolutional network," in [*Proceedings of the IEEE International Conference on Computer Vision*], 576–584 (2015).

[12] Guo, J. and Chao, H., "One-to-many network for visually pleasing compression artifacts reduction," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 3038–3047 (2017).

[13] Kappeler, A., Yoo, S., Dai, Q., and Katsaggelos, A. K., "Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging* **2**(2), 109–122 (2016).

[14] Caballero, J., Ledig, C., Aitken, A., Acosta, A., Totz, J., Wang, Z., and Shi, W., "Real-time video super-resolution with spatio-temporal networks and motion compensation," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 4778–4787 (2017).

[15] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," in [*International Conference on Learning Representations*], (2015).

[16] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going deeper with convolutions," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 1–9 (2015).

[17] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 779–788 (2016).

[18] Girshick, R., "Fast r-cnn," in [*Proceedings of the IEEE international conference on computer vision*], 1440–1448 (2015).

[19] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., and Larochelle, H., "Brain tumor segmentation with deep neural networks," *Medical image analysis* **35**, 18–31 (2017).

[20] Badrinarayanan, V., Kendall, A., and Cipolla, R., "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence* **39**(12), 2481–2495 (2017).

[21] Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., and Carin, L., "Variational autoencoder for deep learning of images, labels and captions," in [*Advances in neural information processing systems*], 2352–2360 (2016).

[22] Beveridge, J. R., Phillips, P. J., Bolme, D. S., Draper, B. A., Givens, G. H., Lui, Y. M., Teli, M. N., Zhang, H., Scruggs, W. T., Bowyer, K. W., et al., "The challenge of face recognition from digital point-and-shoot cameras," in [*2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*], 1–8, IEEE (2013).

[23] Ostermann, J., Bormans, J., List, P., Marpe, D., Narroschke, M., Pereira, F., Stockhammer, T., and Wedi, T., "Video coding with h. 264/avc: tools, performance, and complexity," *IEEE Circuits and Systems magazine* **4**(1), 7–28 (2004).