VISUAL ANONYMITY: AUTOMATED HUMAN FACE BLURRING FOR PRIVACY-PRESERVING DIGITAL VIDEOS

Zihao Liu, Meng Hao, Yuhen Hu

Department of Electrical and Computer Engineering, University of Wisconsin at Madison

ABSTRACT

To conceal the identity of protesters in uploaded video during Arabic Spring demonstrations, Youtube® launched a new tool that allows user to blur human faces in a video before publishing. In this paper, we propose a quantitative model to assess the effectiveness of Youtube's face blurring tool and use this model to guide the development of an alternative algorithm that yields better performance. First, a baseline algorithm is developed using the Harr Cascade Classifier included as part of the OpenCV's object detection module. Leveraging the temporal correlation between successive frames in video sequences, we developed a novel face detection and tracking algorithm. We manually annotated frame by frame four short video sequences and compute the sensitivity and positive predictability of face detection using these three methods. Preliminary results indicate that the proposed new approach significantly reduces the probability of miss detection by OpenCV's baseline algorithm and achieves better false detection rate than Youtube.

Index Terms— Face blurring, face detection, face tracking, Youtube blurring tool, Viola-Jones detectors

1. INTRODUCTION

In July 18, 2012, Youtube launched a new tool that allows user to blur all faces in a video before uploading it to the site. The purpose of this new tool is to protect the identity of protestors of Arab Spring movement so that they are less likely to be arrested by oppressive government agencies based on video clips disseminated on Youtube's human right channel Witness [1]. It turns out that such a capability of visual anonymity has a much wider application such as social networks, news video, and public surveillance.

Because face blurring can be easily branched into face detection, we primarily focus on improving resulting of face detection in this paper. Face detection has been studied extensively over the past few decades. Some approaches focus on developing efficient and accurate machine learning algorithm [3, 4, 5]. The general practice in these approaches is to collect large set of faces and non-faces as training examples and apply certain machine learning algorithm to perform classification of human faces [6]. One of the most popular face detection algorithms based on Adaboost learning algorithm [2], Viola-Jones face detector [3], introduces integral image and cascade classifier on Harr like features that could perform object detection rapidly and achieve high detection rate. Such high rate and accuracy have made Viola Johns face detector widely adopted in the software industry, such as OpenCV's face detection module [7]. Subsequent research has been focused on improvement of the algorithm to tackle factors such as light condition and multi-view face detection [8, 9]. Other approaches concentrate on optimization of face detection through non-learning approaches such as template matching and skin color detection. Skin color can be very useful for detecting faces or non-faces regions in the image [10] and lowering false detection rate [11], while template matching could be used for filtering out the non-face regions [10] to decrease miss detection rate.

In this study, we seek to answer two scientific questions: (a) What is the performance of Youtube's face blurring algorithm (in terms of face detection rates and false detecton rates), and (b) How to further enhance the video face blurring performance if the current algorithm is unsatisfactory.

To address the first question, we have manually annotated four short video sequences to detect the presence and estimate the position of each human face in each frame. These ground truth data allows one to assess performance statistics such as the true positive (TP), false negative (FN), and false positive (FP) rate of a particular face detection algorithm, and derive important statistics such as the sensitivity (*sen*) and positive predictive value (*ppv*).

As a comparison, we developed a baseline face detection and blurring algorithm using Viola-Johns face detector module in the OpenCV software library [reference] to obtain detection and position estimation of human faces in each video frame. We test the performance of this baseline algorithm on the four manually annotated demonstration video sequences and found that the attainable performance is unsatisfactory. In particular, it is observed that faces that are detected in an initially frame often are no longer detected in the subsequent frames after minor pose variations.

To mitigate this shortcoming of the baseline algorithm, we developed a face-detection-tracking algorithm that uses information about detected faces in frame #*n* to predict likely detection and position of faces in frame #*n*+1. Such prediction then will be validated using template matching and color histogram correlation matching. At the present time, this sequential Bayesian like algorithm is implemented as a post-processing step for the baseline algorithm.

With the four video sequences, we have conducted a thorough comparison of the three face detection methods: the Youtube's proprietary algorithm, the baseline algorithm and the proposed face tracking post processing algorithm. Based on the simulation results, it is observed that the proposed algorithm significantly reduces the false negative detection rate of baseline while delivering better FP result. At the present time, the enhanced algorithm is still a bit inferior than that can be achieved by the Youtube algorithm. We propose several approaches to further improve its performance.

The remaining of the paper is organized as follows: background of face detection and blurring are reviewed in Section 2. The baseline algorithm, performance evaluation and proposed enhancement are developed in Section 3; Experiment results and comparison to Youtube is reported in Section 4 with conclusion in Section 5.

2. BACKGROUND

2.1 Face Detection

Face detection has been widely studied in computer vision [3, 6, 8, 9, 10, 11, 12, 13]. Among many competing face detection methods, Viola and Jones [3] proposed a computationally efficient method that uses Haar wavelet like integral image feature, and a cascaded ada-boost pattern classifier to detect the frontal view of human faces in an image. This is also the main algorithm implemented in the OpenCV face detection module [7]. A thorough review of the Viola and Jones' method can be found in [6]. Despite numerous efforts and the proposals of various alternative face detection algorithms, it is observed [6] that in an unconstrained environment, the state of the art face detection algorithms achieve 50-70% correct detection rates at about 0.5-3.0% false positive rates.

2.2 Facial Image Blurring

In addition to the Youtube's efforts, there are also earlier works on face blurring [18, 19, 20]. In [19], two experiments are performed to ask human observers to recognize pixelated and blurred facial images of familiar celebrities. It is reported that participants were still able to recognize some of the viewed faces, despite these image degradations. In addition, moving images of faces were recognized better than static ones. However, for the application of this work, namely protecting the identities of average citizens, this result may not directly applicable. In this work, our focus is on face detection. Thus, once the position and a bounding box of detected face is determined; the content of the bounding box will subject to a low pass filer, resulting in a blurred facial image.

3. RESEARCH METHOD

3.1 The Baseline Algorithm and Performance Evaluation

We employ the face detection module in OpenCV [7] as the baseline face detection algorithm. This algorithm applies the Viola-Jones face detection algorithm on individual frames and provides the location and bounding box of each face detected. Thus, an annotated list of detected faces and their locations in the corresponding frame will be generated.

This face annotation list will be compared to a manually generated "ground truth" list to yield performance statistics. Specifically, three statistics will be accrued: TP (true positive): a detected face is truly a human face at the specified location; FN (false negative): failure to detect a human face at the specified location; and FP (false positive): a detected face does not correspond to a real human face. Another popular statistics TN (true negative) is not available. Based on TP, FN, and FP, two important metrics sensitivity (*sen*) and positive prediction value (*ppv*) can be derived:

$$sen = TP/(TP + FN) \quad (1)$$

$$ppv = TP/(TP+FP) \quad (2)$$

We have collected four video clips with 867, 1036, 1607, and 1292 frames respectively from the Youtube site. We subject these four video sequences to both the baseline face detection algorithm as well as the Youtube face blurring algorithm and compute corresponding sensitivity values and positive prediction values. The results are plotted in Figure 1 below.

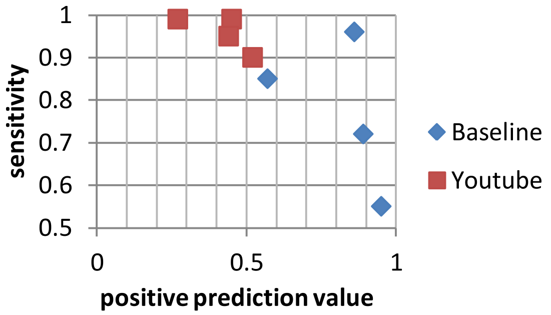

Fig. 1. Performance comparison between baseline and Youtube face blurring algorithms

It is clear that the baseline algorithm is doing fine with ppv values but lags behind in terms of sensitivity. In other words, it fails to detect true faces and hence will not be able to achieve the performance desired. This observation prompted us to seek further enhancement of the baseline algorithm.

3.2 Enhanced Face Detection

To enhance the baseline algorithm, we analyze the annotation results and investigate the cause of its lack of performance. One important finding is that the baseline algorithm has a relative high miss rate in successive frames of the same facial image with slight pose variations. A typical example is shown in Fig. 2 below:

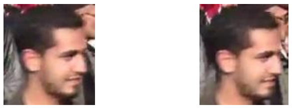

Fig. 2. OpenCV correctly detect first image, but not second, even though two images are consecutive frames.

These missed cases usually happen when the subject rotates his head or the detected region start becoming blurry. We conjecture that this may be due to the baseline algorithm only implemented limited profile face matching templates (the Haar like wavelets) and limited ability to adapt to lighting variations in default setting of threshold parameters. Previous studies of the OpenCV face detection algorithm have led to similar observations [11, 13].

Another important cause for low detection rate is due to the frame-by-frame detection nature of the baseline algorithm. That is, the detection of a human face of the current frame will have no impact on the subsequent frame which is often very similar to the current frame. We hypothesize that ignoring this valuable prior information causes significant performance loss of the baseline algorithm.

Based on these preliminary observations, we propose an enhanced face detection algorithm as a post-processing step after applying the baseline algorithm. A block diagram of this proposed algorithm is depicted in Figure 3 below.

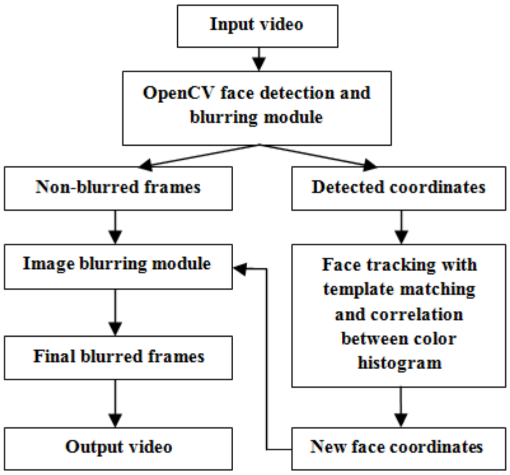

Fig. 3. Block diagram of the proposed system

We initially run OpenCV face detection module to obtain preliminary face detection results. For each detected face in a frame, we examine whether there is also a face detected at neighboring locations in the following frame. If not, we perform face tracking using template matching and color histogram correlation to verify whether there is a similar human face in the vicinity of the location of previously detected face. If both these computed values exceed a preset threshold, we determine there is a new face detected and will include it into the list of detected faces and subject it to the blurred face list. Details of the procedures are summarized in the face tracking algorithm below.

3.3 Face Tracking

The inputs to the enhanced face detection algorithm are the coordination of detected facial images and subsequent frames reporting no face detection at neighboring regions. For each such incidence, the region of interests (ROI) which is the detected facial image of the present frame will be converted into YUV color space. A template matching will be performed at Y component of the ROI at the reference frame seeking best match. Similarly, color histogram will be computed within the ROI and will be compared with similar regions in the neighborhood area in the subsequent frame. In both template matching and color histogram matching, a sample correlation coefficient will be computed according to the formula:

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2}\sqrt{\sum(y_i - \overline{y})^2}} \qquad (3)$$

The search within the frame without face detection will be conducted at a randomly selected displacement from the coordinates of the detected face region. This random shifting approach is to reduce the computation cost of full search motion estimation like procedure.

If the correlation coefficients of both the template matching of gray level component and the color histogram matching of the chromatic components, then it is deemed an undetected face is detected and the position of the best matched ROI will be added to the list of detected faces of that previously undetected frame. We note that this approach is similar to [10, 11, 12] reported previously.

The overall algorithm for face tracking is shown below.

```
Face tracking algorithm
For i = 2, 3 …. n frames do
    Get previous frame as $f_{i-1}$ and current frame as $f_i$
    For each face coordinates $c_i$ in $f_{i-1}$ do
        Random shifting for $c_i$ on $f_i$, get the best shifted coordinate $c_b$ in current frame using correlation on luminance component.
        If chrominance of $c_b$ is also correlated with $c_i$ ‘s chrominance, include it in our improved list
Return improved list
```

3.4 Face tracking improvement

The previous approach works well in terms of increasing the detection rate for the face blurring module. However, one problem we observed is that it also introduces higher rate of false detection. Only correlation between windows in consecutive frames is computed. However, this does not mean that the window contains actual human faces. This problem is illustrated in Figure 4.

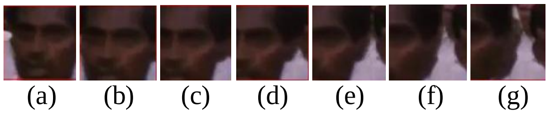

Fig. 4. Sequence of same window using face tracking

After using face tracking method for several consecutive frames, image (a) is very different from image (g). Thus, face tracking could be a very challenging problem. In [17], authors also discuss similar challenges such as illumination variations, pose variations, facial deformations, and occlusion and clutter that would influence the result of face tracking. To improve this situation, we proposed a back tracking method that keeps track of a list of previous windows with faces. Below is the algorithm.

```
Back tracking algorithm
For i = 2, 3 ….  n frames do
    Get a list of coordinates from face tracking module
    For each coordinate
        Check if window is highly correlated to same
        window randomly selected from previous 5th-7th
        frame, if not remove current detection
```

3. EXPERIMENT

To analyze and compare the performance of our algorithm and Youtube's, we developed a system for automatically generating the performance result. We first manually generate coordinates of all real faces in each frame of a sequence of video. This information is our gold standard. Here are criteria we used to decide if we encounter a real human face:

1. Valid faces does not include those with sunglasses but include those with transparent glasses
2. People could still clearly identify the 1 eye, nose and mouth regions on a human face
3. Valid face should be within the orientation of a profile face
4. Valid face should be larger than 25px * 25px in a frame

Examples:

Here are criteria for false positive (not face but detected):

1. The blurring area covers more than a human face is considered as one FP.
2. The blurring area covers the whole picture is counted as five FP.

The result on all test video is shown in Figure 5.

5. DISCUSSION

Our paper suggests a quantitative way to improve the performance of Youtube on blurring real human faces. As shown in the Figure 6, our improved approach overall achieves better result on FP and PPV compared to Youtube and improved TP and Sensitivity compared to baseline algorithm. Youtube does a great job in limiting FN. However, in some cases such as ones shown below in Figure 6., Youtube's blurring technique covers too much in a frame, which is the reason why it has really high FP and low positive predictive value.

	Total faces	TP	FN	FP	PPV	Sensitivity
Video1	760	547 591 683	213 169 77	66 102 633	0.89 0.85 0.52	0.72 0.78 0.90
Video2	2898	2467 2858 2883	431 40 15	1842 1955 3541	0.57 0.59 0.45	0.85 0.99 0.99
Video3	5084	2814 4123 5038	2270 961 46	133 430 6447	0.95 0.90 0.44	0.55 0.81 0.95
Video4	1986	1901 1986 1977	85 0 9	311 986 5438	0.86 0.67 0.27	0.96 1.0 0.99

Fig. 5. Each cell has performance of baseline followed by improved approach followed by Youtube's blurring tool.

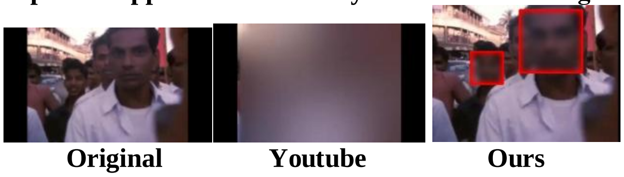

Fig. 6. Examples of comparison with Youtube's

The case presented suggests bad impact on audience as the whole content in the frame is blurred. On the other hand, because Youtube over covers faces, it also achieves highest TP and sensitivity values in all of test videos.

Our improved algorithm is robust to different lighting and poses conditions, as all testing videos contain different orientation of faces and various lighting conditions. The statistics in Figure 5 show that we could achieve similar TP performance compared to Youtube while reducing the FP significantly. However, there are still limitations in our approach. First, our improved algorithm has higher FP than baseline. FP of our algorithm could be further reduced by applying skin color filters on a frame prior to our processing module to filter out the unwanted detected windows [10].

The other limitation of our approach is computational time. It is ideal for offline processing but not for real time face blurring. To compensate for speed limitation, parallel programming principle or client server model should be adopted

6. CONCLUSION

We present an automatic face blurring algorithms for videos that utilizes template matching and correlation between color histograms. The techniques involved are face tracking and improved back tracking algorithm. To efficiently analyze our result, we also developed automatic annotation software that could output annotation result with given gold standard and detected coordinates. Finally, we compare and analyze our result against Youtube and baseline algorithm. Future works need to address problem of improving FP and computational speed.

REFERENCES

[1] D. Netburn, “Youtube’s new face-blurring tool designed to protect activists,”, *Activists,* Los Angeles Times, LA, USA, July. 2012

[2] Y. Freund and R.E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting,” *European Conf. on Computational Learning Theory*, 1994

[3] Paul Viola and Michael Johns, “Robust real-time object detection,” *International Journal of Computer Vision,* 2001

[4] H. Schneiderman. “Learning a restricted Bayesian network for object detection,” *Proc of* CVPR, 2004.

[5] J. Shotton, A. Blake, and R. Cipolla. “Contour-based learning for object detection,” *Proc. of ICCV* ¸2005

[6] Cha Zhang and Zhengyou Zhang, “A survey of recent advances in face detection,” *Microsoft Technical Report*, Microsoft Research, WA, USA, June. 2010

[7] OpenCV, “Harr feature-based cascade classifier for object detection,” *Cascade Classification,* OpenCV, 2009

[8] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang and H. Shum, “Statistical learning of multi-view face detection,” *Proc. Of ECCV,* 2002

[9] B. Wu, H. Ai, C. Huang, and S. Lao. “Fast rotation invariant multi-view face detection based on real adaboost,” *Proc. of IEEE Automatic Face and Gesture Recognition, 2004*

[10] S. Tripathi, V. Sharma, and S. Sharma. “Face detection using combined skin color detection and template matching method,” *International Journal of Computer Applications*, pp. 975-8887, July 2011

[11] M. Zuo, G. Zheng and X. Tu, “Research and improvement of face detection algorithm based on OpenCV,” *International Conference on Information Science and Engineering*, Hangzhou, China, pp. 1413-1416, 2010

[12] L. Wang, T. Tan and W. Hu, “Face tracking using motion-guided dynamic template matching,” *The 5th Asian Conference on Computer Vision,* pp. 23-25, 2002

[13] P. Padilla, C. F. F. Costa Filho and M. G. F. Costa, “Evaluation of haar cascade classifiers designed for face detection,” *World Academy of Science*, 2012

[14] B. Zaman, H., Robinson, P., Oliver, P., and Schroder, H. *Visual Informatics: Bridging Research and Practice*, First International Visual Informatics Conference, pp. 536-537. 2009

[15] J. Maller, “RGB and YUV Color”, *FXScript Reference*

[16] J. Yang and A. Waibel, “A real-time face tracker,” *WACV,* pp. 142 -147, 1996

[17] D. Casas Guix, M. V. Martorell and F. T. Frade, “Real-time face tracking methods,” *ETSE,* June, 2009.

[18] S. Gutta, M. Trajkovic, A. J. Colmenarez, and V. Philomin. "Method and apparatus for automatic face blurring." U.S. Patent 6,959,099, issued October 25, 2005.

[19] K. Lander, V. Bruce, and H. Hill. "Evaluating the effectiveness of pixelation and blurring on masking the identity of familiar faces." Applied Cognitive Psychology 15, no. 1 (2001): 101-116.

[20] I. Mart ńez-Ponte, et al. "Robust Human Face Hiding Ensuring Privacy,” *Proc. Workshop on SemanticMedia Adaptation and Personalization,* Athens, Greece, Dec. 2006.