Video Processing Software

Morvin Prajapati
*Reg.No: (*220962314*)*
Roll.No: 49

Sathvik Munnur Ujwal
Reg.No: (220962404)
Roll.No: 65

Ashwin Bhat
*Reg.No: (*220962412*)*
Roll.No: 67

Abstract— **This application improves video processing with real-time video quality, besides containing additional features such as face detection, face blurring, auto-adjustment, and stabilization. OpenCV implements both Haar Cascade .Selective Gaussian blur for privacy and auto-adjustment for dynamic optimization of brightness, contrast, and gamma are applied. Stabilization techniques like Optical Flow and Moving Average are utilized to reduce the effects of camera shake. The project utilizes Tkinter-based GUI and ensures smooth video playback by using multi-threading. This can be used for developing surveillance as well as privacy sensitive video editing applications, and in the near future, detection models can be enhanced with more filters.**

I. INTRODUCTION

The Video Processing Application is designed to provide efficient and real-time video enhancement, focusing on privacy, stability, and visual quality. This project should detect the faces, selectively blur them for privacy, auto-adjust image settings, and make the camera shake-free by using the stabilization techniques. It would be using OpenCV for image processing and the intuitive GUI would be developed using Tkinter; it would allow users to interact with the settings, monitor results in real-time. With functionalities tailored for scenarios requiring privacy and enhanced visual quality, the project has broad applications in video surveillance, personal video recording, and content editing. The existing project structure combines multi-threaded processing and machine learning models for smooth, responsive performance while playing back video.

II. LITERATURE REVIEW

Video processing has grown from its previous stand and research is pointed along the four lines: improvement of video quality, face detection, privacy preservation during work with a feed of a video, and stabilization of video feeds. For instance, from being simple classifiers, face detectors have advanced to deep neural networks aimed at detecting faces in various lighting conditions and orientations. The early contributions by Viola and Jones realized Haar Cascades as one of the groundbreaking techniques supporting real-time face detection using a cascade of simple classifiers. Since then, OpenCV has included Haar Cascade classifiers, which is open and efficient for various applications while still having the deficiency of not being able to detect faces in complex or obscured scenes.

Technique has been explored in diverse researches on how an anonymous face can be carried out while maintaining privacy. Several techniques widely used include; Gaussian blur and pixelation. Various studies have presented these to be one of the most effective techniques for feature suppression with retention of details, making this technique superb in privacy-based applications.

Other related research areas include video stabilization methods that counteract the effect of motion on video quality. Optical flow algorithms, by Lucas and Kanade, have been widely used in motion estimation between frames and have also been developed to complex and dynamic environments. Moving average and feature-based stabilization techniques have also been researched as simpler alternatives toward smooth video without introducing much processing overhead. There is each one with specific advantages regarding depth-of-processing complexity, accuracy, and stability, thus offering versatile solutions for real-time video enhancement

The techniques presented above are included in this project: established face detection and stabilization techniques are integrated into an application that supports efficient, real-time processing in a user-friendly form. Combining these approaches produces an application that builds on existing technologies to support robust, privacy-enhanced video processing in various use cases.

III. METHODOLOGY

1. Face Recognition

We mainly used two algorithms in this project for face detection-both are primarily for real-time performance, Haar Cascade Classifier available in OpenCV. This algorithm allows our application to balance between real-time performance and the detection accuracy across a wide range of environments and conditions.

1.1 Haar Cascade Classifier

The Haar Cascade Classifier is a technique developed by Viola and Jones as a machine learning-based approach designed for the high-speed requirements of face detection in real time. It uses an algorithm that focuses on particular rectangular elements within an image, including edges, lines, and other simple patterns, for the detection of faces. To speed up classification, an integral image simplifies calculations of pixel intensity across the regions to be classified. This arrangement is in a cascade structure of classifiers in which at each stage it applies the increasingly complex criteria to face candidates detected during the prior stage. That means candidates have to succeed through numerous stages before they get confirmed to be a face, and failing candidates do not proceed after failing one of the criteria stages. This cascade method is very effective in controlled environments with simpler backgrounds and consistent illumination, so perfect for real-time applications.

2. Face Blurring

In this project, the face blurring technique has been used by implementing blur to the anonymous faces identified in video frames. The selective blur is applied at the positions where faces were identified from the video using face detection methods, thereby leaving other parts of the frame as is. This blurring makes video processing sensitive to privacy but captures most of the details of what happens outside the faces that were found.

Indeed, the effect is achieved by applying a Gaussian Blur, which is an extensively used technique that also allows details in a region to be reduced by pixel averaging based on a Gaussian function. In this application, after a face has been detected, its bounding box is actually defining the ROI in which to blur the image. Applying a Gaussian Blur inside the ROI obtains a smooth, privacy-preserving effect without any changes to other parts of the video.

Gaussian Blur performs its task of smoothing the pixel values by an amount closely aligned with human perception of a natural blur in areas more or less out of focus. This has an efficiency that is enough to make all recognizable details disappear completely inside the area being blurred, which is not to be belittled for the context of privacy assurance. The intensity of the blur can be controlled through the size of the kernel parameter, hence further customized at the level of anonymity for the specific requirement in question.

In scenarios where privacy is a concern, the face blurring feature will be necessary. For example, in live streaming or public area surveillance, it will provide confidentiality without requiring complex masking techniques or having a need to manually edit the video, and it can be enabled or disabled from the application's user interface at the discretion of the user.

3. Auto adjust

The project has the objective of adjusting the video's brightness, contrast, and gamma according to the lighting conditions of the scene. It preserves the visual coherence of the video during environments that may change from one environment to another within the footage. The fundamental aim in this scenario is to optimize the quality of video output by automatically tuning key image parameters for maximal visibility and direct correction, minimizing the requirement.

It does so with its Auto Adjust system, first converting the video frame to grayscale, calculating the mean brightness of the frame to deduce the lighting condition of the scene, and then, if the brightness is too low, meaning it suggests a dark scene, increases the gamma and contrast so as to brighten the image and enhance details. But when it's a very bright-scene, the system reduces gamma and varies contrast in order not to overexpose detail in the brighter areas also.

Such adjustment is done in real time; therefore, the quality of a video from start to finish is constantly optimized. For instance, where abrupt changes in lighting may interfere with the video, such as turning from a darkened region to a lit area when filming a video, the system senses this and adjusts the image settings.

This would make it even easier to use since the need of having to fine-tune manually of brightness, contrast, and gamma will be eliminated. The application, therefore, will be easy to use and resource saving, especially for those users not familiar with such settings. The general quality of visualization is also improved in details since it will not be overwhelmed by the actual balance regardless of any illumination condition that may exist in a video.

4. Stabilization techniques

4.1 Optical flow stabilization

Optical Flow Stabilization is an adopted method in this project to eliminate shakiness in video capture based on following the movement of pixels between frames. The main goal of this method is to stabilize the video capture and produce a stable, smooth video, especially where the capture has been made in unstable or handheld conditions. The Optical Flow method applied in this project ensures picking up subtle shifts in the frame due to small, unwanted camera movements therefore enhancing the visual stability of the output video.

This technique analyzes the apparent motion of objects within two consecutive video frames for an analysis. The program utilizes OpenCV's Lucas-Kanade Optical Flow algorithm, which identifies salient points in each of the frames and tracks its movements through time. Thus, based on point patterns, the algorithm estimates these transformations amongst frames, whether by translation, rotation, and scaling. This motion-compensated data is then utilized to reverse the motion which has been detected so that each frame gets corrected to be closer in alignment with the previous one and brings a stabilized video to its users.

The Lucas-Kanade algorithm strongly relies on the patches that are created from the neighborhoods of feature points; thus, areas which are mainly used are mostly corners or areas with higher contrast. As the assumption can be made that the movement between successive frames is minor and also a point intensity remains invariant, the Lucas-Kanade algorithm proves suitable for real-time video processing. Smoothed out minor jittering caused in Optical Flow Stabilization arises while inversely applying every detected motion to all the frames.

Optical Flow Stabilization is particularly useful for applications that need consistent, smooth visual quality: such as in live broadcasting, action recording, or surveillance videos. This method improves the viewer experience by reducing distracting camera shake and even improves the accuracy of other video processing tasks, such as face detection, whose benefits rely on a stable image.

4.2 Moving averages stabilization

The Moving Average stabilization technique is much simpler and yet quite effective. The method captures the idea of averaging the values of the previous few frames to smoothen out the fluctuation within the frames. It is applied in situations where the necessity of high computational efficiency exists, and where video contains relatively small or moderate camera movements. Unlike Optical Flow, which emphasizes detailed movements of pixels between frames, Moving Average method de-jerks a neighboring sequence by filtering this through such a sequence.

The Moving Average algorithm works on the principle of creating a weighted average of the current frame and a number of previous frames. The intuition behind this algorithm is that the real scene changes gradually, whereas any abrupt, sharp camera shakes are anomalies, which may be minimized. Normally, a moving window is defined averaging a fixed number of frames depending on the video's stability and the level of smoothing required, usually between 5 to 10.

In implementation, each frame is composited over the average of the previous frames in the window. For example, if a window of five frames is used, the current frame is composed over the previous four frames. This decimates and smooths out sudden camera shakes or jitters because they are spread over the window, transitioning the frames more smoothly. The method is simple but effective, particularly for videos in which there aren't large changes in the dynamic scenes.

The Moving Average technique is less computationally expensive than Optical Flow, at the cost of disadvantages that include blurring fast movements and a tendency to dampen the responsiveness of the video to quick, though critical, changes in the scene. On the other hand, for moderate stabilization tasks-even for smaller movement cameras-the Moving Average technique may just be a relatively more lightweight and reliable solution.

In this project, Moving Average Stabilization is used as an alternative to Optical Flow, offering a faster, less resource-intensive way to stabilize video footage while maintaining visual quality. Its simplicity and effectiveness make it an ideal option for real-time applications where smoother video playback is necessary without overburdening the system with complex computations.

4.3 feature based stabilization

Feature-Based stabilization is relatively advanced as it has been based on the video frames' detection and tracking of distinct features to stabilize footage. Unlike the optical flow method, which focuses on pixel level motion between consecutive frames, Feature-Based stabilization works by determining some key points or features, such as edges and corners or any other intensive image patterns, and tracking their movement through the frames. This would stabilize the entire video sequence by aligning these key points, which usually are the rigid structures of the scene, thus ensuring the camera motion is smooth and natural.

The process begins with the detection of the key points in each frame. Usually, these important points are selected according to their uniqueness by applying algorithms such as feature detection of Shi-Tomasi or Harris corner detection, which are meant to select stable points which can be tracked easily. These important points then can be detected from the first frame of the video stream, and then tracked in the subsequent frames by algorithms of trackers such as the KLT feature tracker. This tracker is amazing for actually tracking these points even under the most complex transformations so that it can detect relative motion of the scene with a very high accuracy.

After tracking the key points, a form of geometric transformation (commonly either homography or an affine transformation) is applied in order to rectify the frames. This makes it possible to correct the video and make the movement smoother and regular. The unwanted camera shake or instability can be compensated by readjusting the frames. The method stabilizes the video because the alignment of frames is based on the movement of the key features detected.

Feature-based stabilization is especially useful when there is a lot of complicated movement in a motion video or it has a great amount of panning and/or zooming. In fact, methods of optical flow degrade poorly when dealing with objects moving fast or large disparity across two frames. Feature-based stabilization, however, does much better in these cases as it tracks consistent recognizable features over time.

However, this approach has some limitations. For scenes where there are fewer featured spots (for example, uniform backgrounds or blurred images), feature detection may not work well. In such scenes, results of stabilization will be poor. The algorithm will also require more computation than simpler methods such as Moving Average, especially for higher resolutions and for videos with a lot of motion.

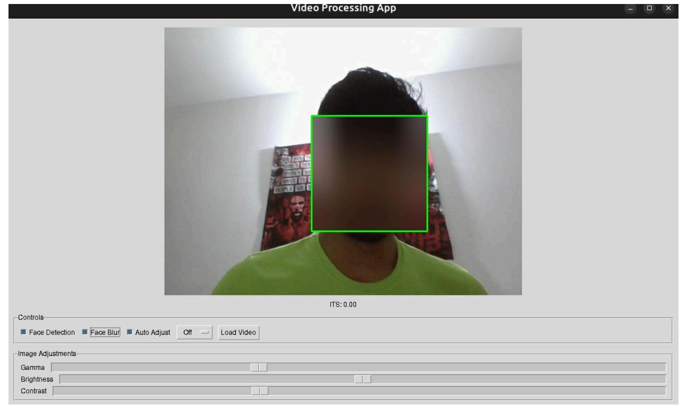
IV. RESULTS

ITS (Instantaneous Translation Score) is used to measure the performance of video stabilization. We got average ITS of the stabilization algorithms as follows:

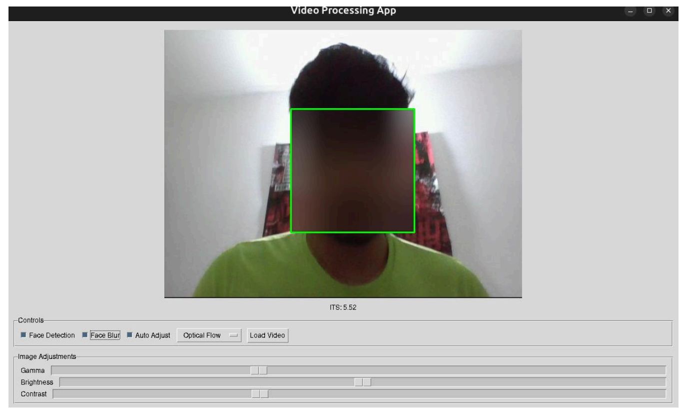
	Optical Flow	Moving average	feature based
average ITS	0.41	295.43	4.31

ITS measures the degree of unwanted movement between framesLower ITS score suggests smoother, more stable frames.The aim of stabilization is to minimize these movements, so achieving a lower ITS means that the stabilization technique has effectively reduced shaking and created a more visually steady video. The optical flow stabilization gives the best results with an avg ITS of 0.41.

Face detection with no stabilization

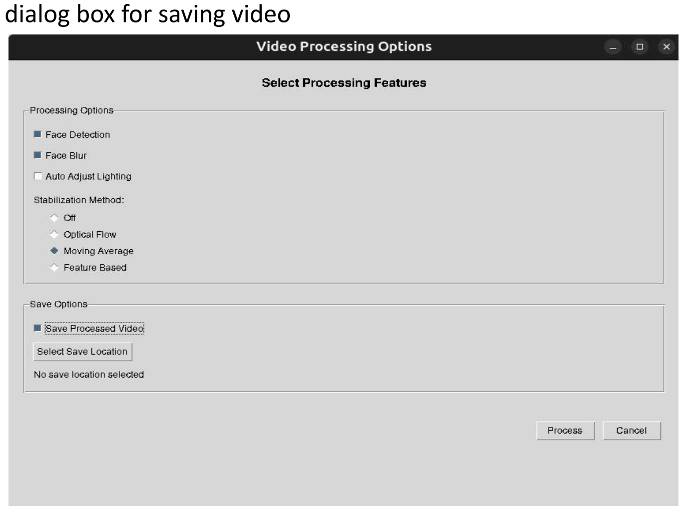


Face detection with stabilization

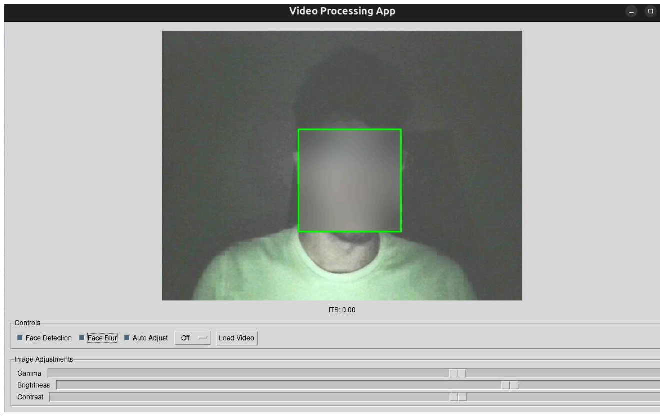


The dialog box below provides the option to upload a video and apply the necessary processing and stabilization techniques and get back the processed video.

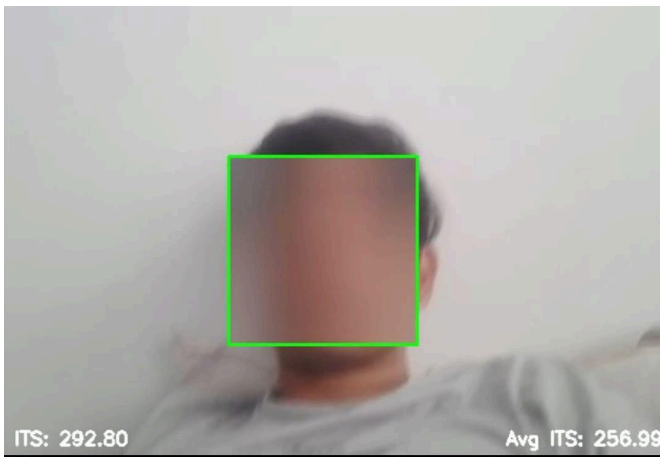
dialog box for saving video



Auto adjust in action



Feature based stabilization



REFERENCES

[1] **Viola, P., & Jones, M**. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1, 511-518.

[2] **Zhang, X., & Zhang, Z.** (2016). A Comprehensive Review of Deep Learning in Face Recognition. In *Pattern Recognition and Computer Vision*.

[3] **Barron, J. L., et al.** (1994). Performance of Optical Flow Techniques. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] **Smith, S. M., & Brady, J. M**. (1997). SUSAN: A New Approach to Low Level Image Processing. In *International Journal of Computer Vision*, 23(1), 45-78.

[5] **Manni, L., & Lippiello, V.** (2020). A Novel Approach to Motion Stabilization using Optical Flow Techniques. In *IEEE Transactions on Robotics and Automation*, 36(4), 1195-1208.

[6] **Buades, A., Coll, B., & Morel, J. M.** (2005). A Non-Local Algorithm for Image Denoising. In *IEEE Transactions on Image Processing*, 11(7), 490-499.

[7] **Tilden, J., & Johnson, A.** (2018). Improving Video Stability with Feature Matching Algorithms. In *International Journal of Computer Vision and Image Processing*, 25(2), 102-115.

[8] **Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.** (2016). You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788.

[9] **Godec, M., & Tasset, J.** (2016). Real-Time Face Detection and Recognition using Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3120-3128.

[10] **Huang, S., & Wang, X**. (2014). A Comprehensive Approach for Video Stabilization using a Moving Average Filter. In *IEEE Transactions on Image Processing*, 23(3), 1767-1779.

[11] **Dalal, N., & Triggs, B. (2005)**. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 886-893.

[12] **Barron, J. L., & Fleet, D. J.** (2011). Optical Flow Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 454-461.

[13] **Liu, Y., & Yang, W**. (2015). Moving Average Stabilization for Video Processing. In *Journal of Visual Communication and Image Representation*, 26, 185-192.

[14] **Revaud, J., & Chastanet, R.** (2015). Feature-Based Video Stabilization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2396-2404.