Module 8: Portfolio Project

Brady Chin

Colorado State University Global

CSC515-1: Foundations of Computer Vision

Dr. Bingdong Li

May 11th, 2025

Portfolio Project

The purpose of his project is to identify human faces and blur their eyes for various images. The program uses OpenCV's pre-trained Haar cascade classifiers to process images to identify facial elements so that privacy can be preserved by obscuring identifiable facial features. The processing of images present a number of different image aspects including lighting, distance of subject to camera, and potential for more than one person in the same image, and this report details the chosen detection, preprocessing, and parameter variations for accuracy, plus reflections on aspects of the programming process faced through to implementation of results.

Techniques Used

In this project, Haar cascade classifiers were the primary method used. Haar cascade classifiers are a method common to object detection within the OpenCV library. The pre-trained models for paper face and eye detection were loaded within OpenCV with the help of the CascadeClassifier class (su77ungr 2022, November 17). Haar cascade classifiers classify objects by scanning the image at several scales and locations then, for every candidate image, the classifier classifies the image as a feature or non-feature based on any features the classifier remembers from training (Mittal, A., 2020, December 20). The reason for Haar cascade classifiers was the simplicity, speed, and existence in OpenCV. Therefore, they were an appropriate, though basic, detection tool for facial features.

To improve the classifiers' detection success, the input images had been preprocessed first before running the classifiers. The preprocessing of each image included, converting images to grayscale, equalizing the histogram to enhance contrast among the face images, and using a bilateral mean filter to minimize noise and preserving edges. Preprocessing helped to enable the classifiers to identify facial features with poor lighting conditions and inconsistent image quality.

First, I performed face detection to detect the faces in each photo. After the faces were detected, I proceeded to eye detection for every detected face. After performing eye detection, the model would blur the eyes with a Gaussian blur function to hide the faces or other identifying features. We used Gaussian to obscure the eyes because it lends a natural smoothness and we preferred blurry details where there were no hard artifacts.

At first, the faces and eyes either did not detect properly or it was detecting incorrect features. To fine tune, I made changes to the parameters of the detectMultiScale function (scaleFactor, minNeighbors, and minSize). These parameters set how the classifier scales through the image, how many neighbor detections have to occur for an object to identified, and the minimum possible size for the object to be detected (OpenCV Documentation., 2025).

Challenges

One major challenge for this project was the inability to consistently detect faces and eyes amongst images. The Haar cascade classifiers had reasonable detectability with clearly visible, full-frontal images; however, the detectability definitely decreased once there were multiple people, subjects at greater distances, and different skin tones. The classifier sometimes missed some of the faces, and it even recognized parts of non-face areas as eyes. Because of this, every image had inconsistent detection outcomes.

The other challenge was setting the detectMultiScale parameters. There was a fine line in detecting all the features while also avoiding detections that were not actual true positives. If the minNeighbors value was decreased, there will be more detections but many of which were false positives. On the other hand, if minNeighbors was increased, it would become more difficult to detect the correct features. Figuring out viable scaleFactor and minSize numbers took some trial and error particularly when looking for eyes

Results

After some trial and error, I was able to detect the faces and eyes and blur the eyes for the 5 chosen images. I had the most difficulty with the multiple people image and the image from a distance. For the image with multiple people, this was mostly caused because of the different skin tones between each person. I achieved better results when all people were similar skin tones. The far away image was especially difficult since the eye details were so small. I was able to detect the eyes only when the person was looking directly at the camera for this image.

Conclusion

The project achieved the goal of detecting human faces and obscuring their eyes using Haar cascade classifiers specifically pre-trained models inside of OpenCV. While challenges were faced with detection accuracy across different images and even, different parameters to tune, after enough iterations, a satisfactory outcome was achieved. The use of preprocessing was especially beneficial since detection accuracy regarding perturbation can greatly differ if preprocessing methods such as histogram equalization and bilateral filtering are used. The project has demonstrated the importance preprocessing and optimizing relevant parameters has in relation to object detection tasks, and provided deeper insights into detecting a real-world image's potential obstacles. While Haar cascades may never achieve the alleged cloak of invisibility in face and feature detection, their time and computational efficiency means that they will more than suffice for face and eye detection.

References

Mittal, A., (2020, December 20), *Haar Cascades, Explained.* Medium. https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d

OpenCV Documentation, (2025), *cv::CascadeClassifier Class Reference.* OpenCV. https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html

su77ungr (2022, November 17), *OpenCV.* Github. https://github.com/opencv/opencv/tree/master/data/haarcascades