Face and Eye Detection with Blurring Using OpenCV – Option2

Submitted by - Arun Kumar Saxena

Prof – Jonathan Vanover

Date – 11/03/2024

# Introduction

The purpose of this project is to develop a Python application that uses OpenCV for detecting faces and eyes in images and applies blurring to the detected eyes. This process involves using Haar Cascade classifiers for face and eye detection and applying image preprocessing techniques such as histogram equalization for enhancing the detection accuracy.

# Code Implementation

The project is implemented in Python, utilizing the OpenCV library for image processing tasks. The main functionalities include:

* Loading and preprocessing images
* Detecting faces in images using Haar Cascade classifiers
* Detecting and selectively blurring eyes within the detected faces

## Dependencies

This project requires the following dependencies:

* Python 3.x
* OpenCV library (cv2)

## Code

A screenshot of a computer code

Description automatically generated

# Explanation

## Face and Eye Detection

The project uses Haar Cascade classifiers for detecting faces and eyes. These classifiers are pre-trained and provided by OpenCV. The `cv2.CascadeClassifier` function is used to load the classifiers for faces and eyes.

## Image Preprocessing

The `preprocess\_image` function converts the image to grayscale and applies histogram equalization, which enhances the contrast in the image, making it easier to detect features such as faces and eyes.

## Blurring Eyes

Once faces are detected in the image, the code isolates the region of each face and then detects eyes within that region. The detected eyes are then blurred using Gaussian blur to anonymize or obfuscate them. This is achieved by replacing the eye region in the image with its blurred version.

## Testing and Results

The application was tested with three different types of images:

* Multiple faces in a single image: The application produced good results but had some false positives and negatives.
* Non-human image of a monkey: The application incorrectly identified the monkey's ears as faces.
* Full body image: The application was able to clearly identify the face in the image.

By experimenting with different scale factors, I was able to resolve some of these issues and improve detection accuracy.

## Displaying Results

Blurring faces -

A collage of people's faces

Description automatically generated

Non human

A close up of a monkey

Description automatically generated

Full Image

A person in a black dress

Description automatically generated

# Challenges and Observed Results

1. **False Positives**:
   * In a non-human subject image (monkey), the algorithm incorrectly detected ears as faces. Despite parameter adjustments, the algorithm struggled to fully distinguish human faces from similar shapes in non-human subjects, leading to false positives.
2. **False Negatives**:
   * In the full-body image, the face was detected successfully; however, certain images yielded false negatives or missed detecting faces in distant or smaller faces. Tuning the parameters improved accuracy for closer faces but didn’t entirely resolve detection issues for far-away subjects.
3. **Lighting and Contrast Variations**:
   * Adjusting the contrast via histogram equalization improved feature visibility, particularly in low-light images. This step enhanced detection but wasn’t able to fully eliminate misdetections due to challenging lighting.
4. **Eye Detection Challenges**:
   * Glasses or obstructions on subjects' faces reduced eye detection accuracy. Although parameter tuning helped, this solution remains somewhat limited for such cases, as more advanced models would be required for complete robustness.

# Conclusion

This project demonstrates the use of OpenCV for face and eye detection in images, along with the application of blurring to the detected eyes. The implementation highlights the capabilities of image preprocessing and feature detection using Haar Cascade classifiers. The results are visualized by displaying the processed images with the detected faces and blurred eyes.