Face Blurring

Methodology

Face Detection

The first step involves locating faces in an image. For this purpose, the YuNet face detection model was employed due to its high accuracy in detecting faces under varying lighting conditions, sizes, and positions.

Face Blurring

After identifying bounding boxes around detected faces, the next step is to blur them to obscure identities. Two blurring techniques were explored: Gaussian Blur and Pixelization. Pixelization was ultimately chosen for its more aesthetically pleasing results, as Gaussian Blur produced less visually appealing outputs.

Pixelization Process

Pixelization is an image processing technique where pixels in an image are replaced with larger blocks of a single color. This is achieved by dividing the image into a grid of cells, where each cell corresponds to a group of pixels. The color or intensity of all pixels in the cell is replaced with the average value of the colors in that cell. This process simplifies the image, reducing fine details and making it difficult to identify individuals. The degree of blurring is controlled by the size of the pixel blocks.

Implementation Steps

- **Image Preprocessing**: The input image's dimensions (height and width) are extracted, and the YuNet detector is configured to match the image size for optimal face detection.
- **Face Detection:** The model processes the image to identify faces and extracts their bounding box coordinates (x, y, width, height). These coordinates are validated to ensure they fall within the image boundaries.
- **Pixelation Process:**
    1. The identified face regions are extracted.
    2. Each face region is resized to a very low resolution (e.g., 10x10 pixels) using nearest-neighbor interpolation.
    3. The low-resolution version is scaled back to its original size, creating a pixelated effect.

4. The pixelated face replaces the original face in the image.

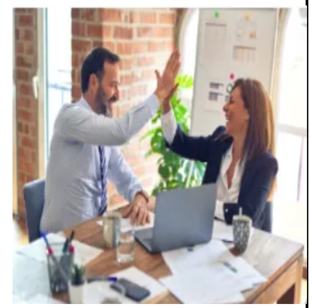
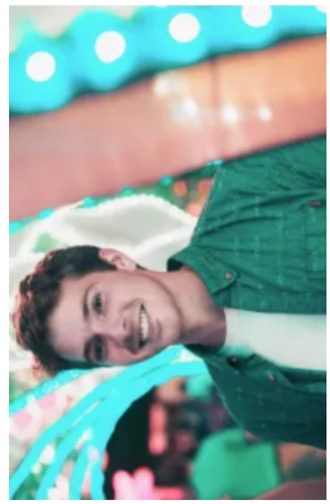
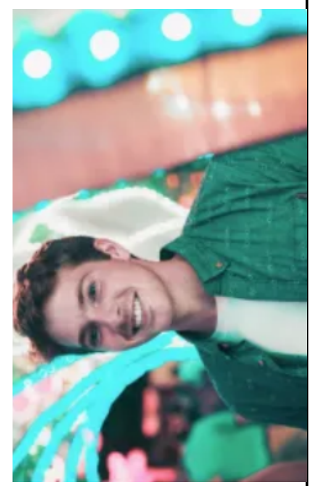
Results and discussion

The algorithm was evaluated using several test images to assess its performance under various scenarios, including:

1. Group of people close together: To test the algorithm's ability to detect faces of different sizes and distances.
2. Side-view faces: To evaluate the detection of faces not directly facing the camera.
3. Flipped faces (90 degrees): To assess the algorithm's capability to detect rotated faces.

The results were compared with other models, including YOLOv5, MTCNN. As shown in the table below, YuNet outperformed the other models in all scenarios, particularly in detecting faces of varying sizes, orientations, and skin tones.

Original Picture	Our Model (YuNet)	YOLOv5	MTCCN
			
			

Conclusion

As shown in the table above, our model outperformed the other three models. MTCNN exhibited bias against individuals with darker skin tones and struggled to detect faces that were not directly facing the camera. It also performed poorly on rotated images and images containing multiple or distant faces. Similarly, YOLOv5 had difficulty detecting rotated faces and failed to recognize a woman's face from a side angle. In contrast, our model successfully detected faces across all these challenging scenarios, demonstrating superior performance and robustness.