The Impact of Image Pre-processing on Face Detection in Poor-Quality Images

Bhavana mittapalli-24114
Ganesh Guguloth - 24030

1 Introduction

This project investigates the impact of image pre-processing techniques on improving the performance of face detection algorithms in poor-quality images. Image pre-processing aims to enhance the quality of the input image by reducing noise, improving contrast, and mitigating blur, thereby providing a clearer and more consistent basis for face detection. The main goal of this study is to evaluate how various pre-processing techniques—such as denoising, contrast enhancement, and deblurring—affect the accuracy and robustness of different face detection methods.

The face detection algorithms under consideration in this study span both traditional and modern approaches. Traditional algorithms, such as the Viola-Jones (Haar Cascade) classifier, have long been used in real-time face detection due to their speed and efficiency. On the other hand, modern deep learning-based methods, such as Multi-task Cascaded Convolutional Networks (MTCNN) and You Only Look Once (YOLO), offer higher accuracy, particularly in challenging conditions. By evaluating the performance of these algorithms with and without pre-processing techniques, we aim to determine the extent to which pre-processing can improve detection outcomes.

Ultimately, this project aims to provide valuable insights into the effectiveness of image pre-processing in enhancing face detection capabilities and whether traditional face detection methods benefit more from these techniques compared to modern deep learning-based models. By understanding the impact of these techniques, we can improve face detection in real-world applications where image quality is often less than optimal.

2 Face Detection Algorithms

2.1 Viola-Jones (Haar Cascade)

Haar Cascade: Traditional algorithm based on Haar-like features and a cascade of classifiers. Known for real-time performance but limited robustness to variations in image quality. The key steps involved are:

- **Feature Extraction**: Haar-like features are computed using the integral image, capturing patterns such as edges, lines, and textures within rectangular regions.
- **Feature Selection**: AdaBoost is used to select the most discriminative features that separate faces from non-faces.

- **Cascade of Classifiers**: A series of classifiers eliminates non-face regions quickly, focusing computational effort on promising areas.

- **Multi-scale Sliding Window**: A detection window of varying sizes scans the image to detect faces at different scales.

- **Bounding Box Refinement**: Detected bounding boxes are refined and merged to produce the final results.

2.2 MTCNN:

Multi-task Cascaded Convolutional Networks. Combines face detection and facial landmark localization, offering strong performance in challenging conditions. The key steps are:

- **Model Initialization**: MTCNN was initialized with the `keep_all=True` parameter to detect multiple faces in an image. Device selection (CPU or GPU) ensured efficient execution.

- **Face Detection Workflow**: Input images were converted from BGR to RGB format. The `mtcnn.detect` method was used to identify bounding boxes for faces, along with confidence scores to assess detection reliability.

- **Algorithmic Advantages**: MTCNN uses a cascaded structure of three neural networks:
  - **P-Net**: Proposes candidate face regions.
  - **R-Net**: Refines these candidate regions.
  - **O-Net**: Produces final bounding boxes and, optionally, facial landmarks.

- **Output**: The number of detected faces was reported, and annotated images with bounding boxes and confidence scores were displayed.

3 Pre-processing Techniques

3.1 Deblurring:

In the deblurring step, we employed two techniques: Blind Deconvolution and Wiener Deconvolution. The parameters were tuned as follows:

Blind Deconvolution: For Blind Deconvolution, we tested a range of kernel sizes to evaluate their effect on the quality of the restored images. The kernel sizes used were:

- 3×3

- 5×5

- 7×7

These kernel sizes represent increasing levels of complexity in the deblurring process. By varying these sizes, we were able to assess how the amount of information used in the restoration process impacted the accuracy of face detection.

Wiener Deconvolution: In Wiener Deconvolution, two key parameters were tuned: the kernel size and the noise variance. The kernel sizes tested were:

- 3×3
- 5×5
- 7×7

Additionally, the noise variance parameter was varied to examine its effect on the deblurring process. The two values used for noise variance were:

- 0.01
- 0.02

By adjusting the noise variance, we explored how different levels of noise suppression influenced the effectiveness of the deblurring technique and the overall face detection performance.

3.2 Denoising:

—For the denoising step, we applied two distinct techniques: Gaussian Filtering and Total Variation (TV) Smoothing. The parameters for each method were carefully tuned to evaluate their impact on noise reduction while preserving key facial features for improved face detection. The tuning process for each technique is described below:

Gaussian Filtering In the Gaussian Filtering method, we experimented with varying kernel sizes and sigma values to optimize the level of smoothing applied to the images. The following parameters were used:

- **Kernel Sizes**:
  - 7×7
  - 9×9
  - 11×11

  These kernel sizes were chosen to examine the effects of different levels of spatial smoothing on the noise reduction process.

- **Sigma Values**:
  - 1.5
  - 2
  - 2.5

  The sigma values were varied to control the spread of the Gaussian kernel, which influences the strength of the smoothing effect. A higher sigma results in a broader kernel, leading to more aggressive noise removal but possibly blurring fine details.

By adjusting both the kernel size and sigma value, we aimed to determine the optimal settings for minimizing noise without excessively blurring important facial features.

Total Variation (TV) Smoothing For Total Variation (TV) Smoothing, we tuned the weight parameter, which controls the balance between noise reduction and edge preservation. The following weight values were tested:

- 0.2
- 0.5
- 0.8

3.3 Contrast Enhancement

For contrast enhancement, we applied the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique. In this method, we varied two key parameters: the clip limit and the tile size. The goal was to adjust these parameters to optimize contrast improvement without introducing artifacts. The parameters were tuned as follows:

CLAHE (Contrast Limited Adaptive Histogram Equalization) The following parameters were varied to assess their effect on the image contrast enhancement:

- **Clip Limit**:
  - 2
  - 4
  - 6

  The clip limit parameter controls the intensity limit for contrast enhancement. By adjusting the clip limit, we aimed to control how aggressively the contrast was amplified in the image. A higher clip limit enhances contrast more significantly, but may result in visible artifacts such as noise amplification.

- **Tile Size**:
  - 8×8
  - 10×10

  The tile size determines the local region of the image where histogram equalization is applied. Smaller tiles provide finer adjustments to local contrast, while larger tiles result in more global adjustments, leading to smoother results. We varied the tile size to evaluate the balance between local and global contrast enhancement.

Through this parameter tuning, we aimed to determine the optimal combination of clip limit and tile size for enhancing the visibility of faces without introducing undesirable noise or artifacts.

4 Dataset

The dataset consists of **40 original images** containing faces, with varying poses, expressions, and lighting conditions. These images were degraded into three versions to simulate real-world challenges:

- **Blurry**: Introduced motion blur and defocus.
- **Noisy**: Added Gaussian noise to simulate low-light conditions.
- **Low-Contrast**: Reduced contrast to mimic poor lighting.

Each degraded version underwent **pre-processing techniques** to improve quality:

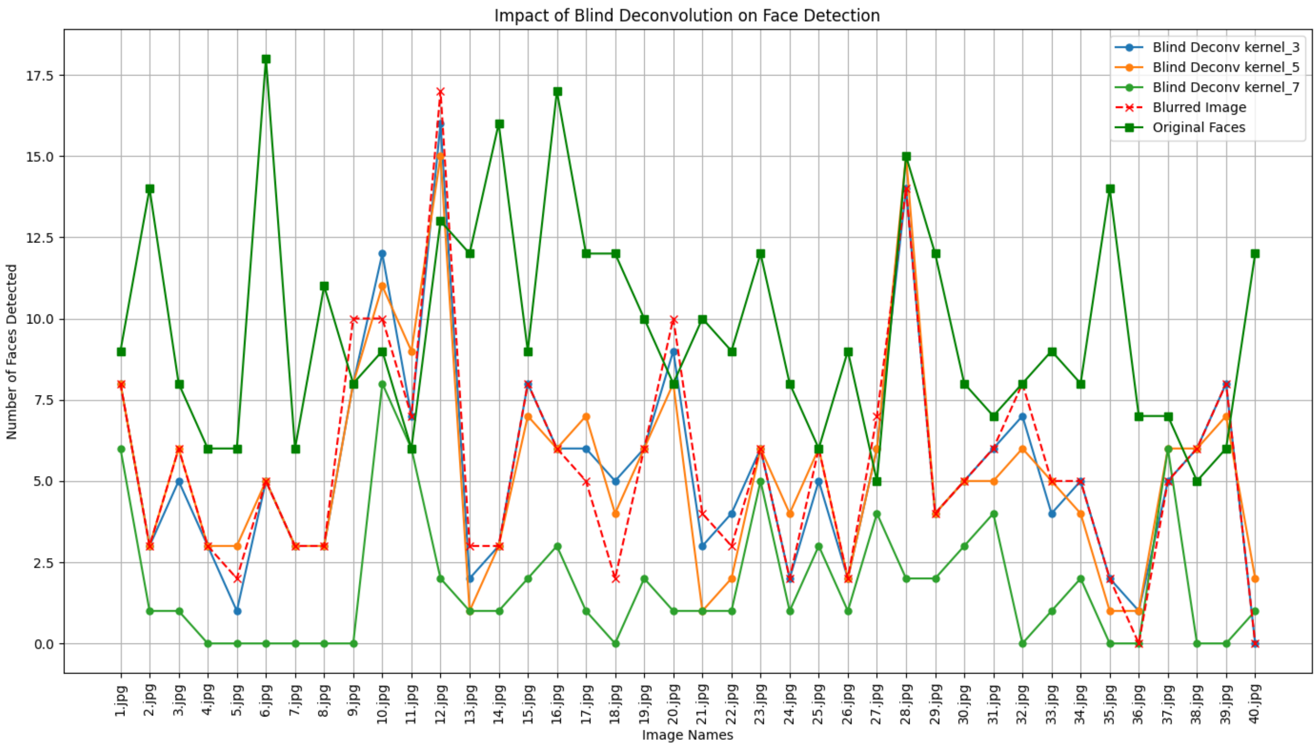
- **Deblurring**: Applied **Blind Deconvolution** and **Wiener Deconvolution**.
- **Denoising**: Used **Gaussian Filtering** and **TV Smoothing**.
- **Contrast Enhancement**: Employed **CLAHE**.

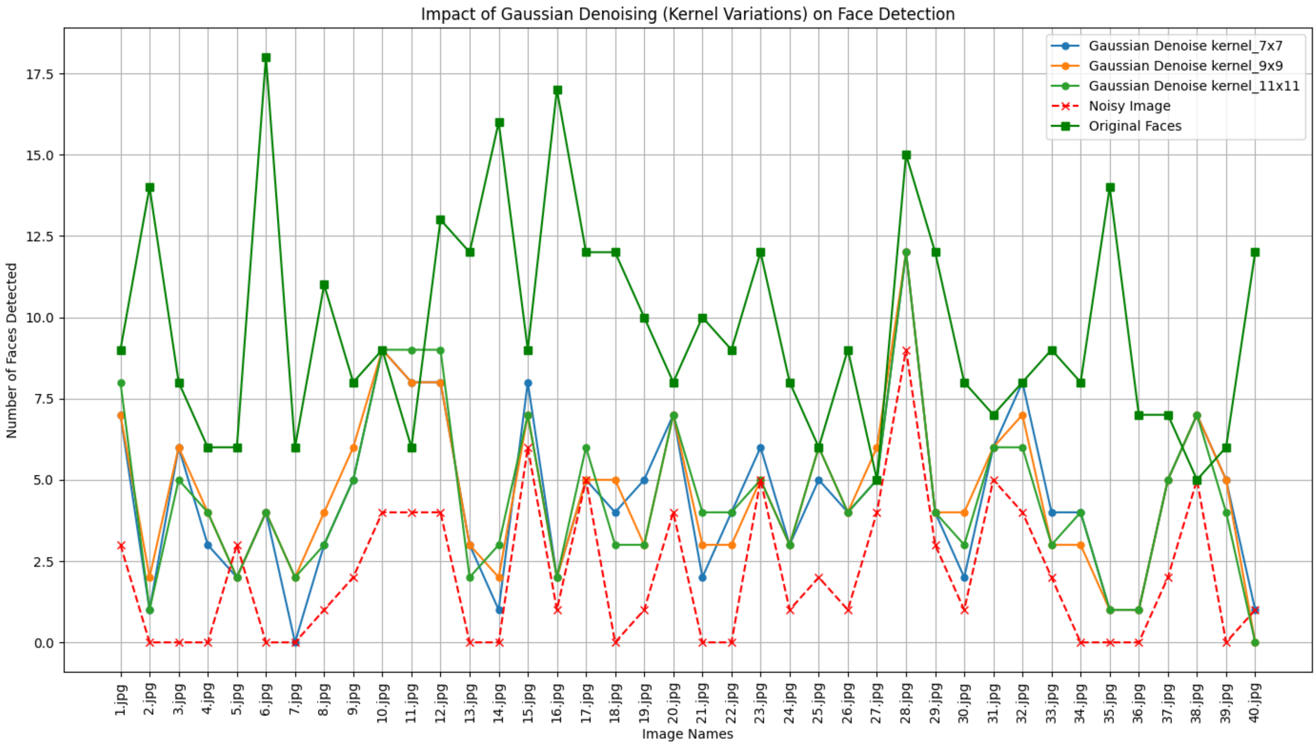
All four image sets (original, blurred, noisy, low-contrast) were tested with **Haar Cascade** and **MTCNN** face detection algorithms to evaluate the impact of the pre-processing techniques on detection performance.

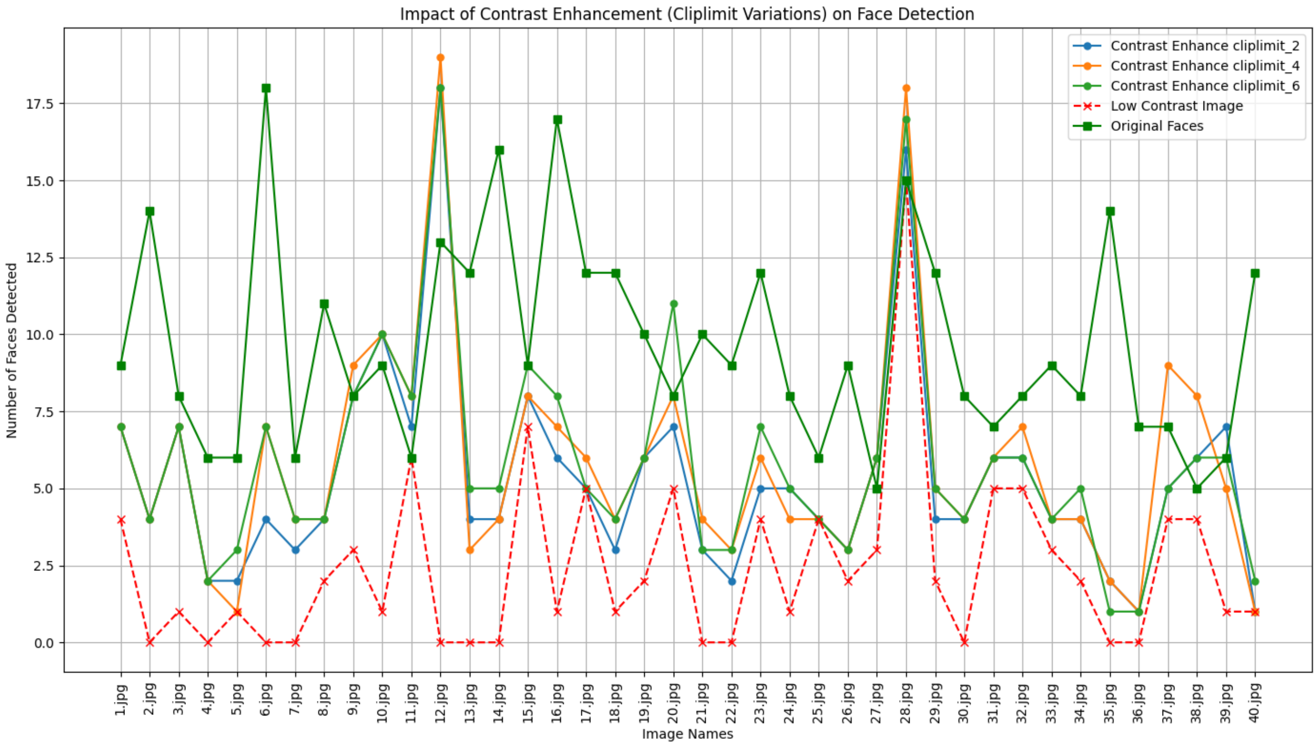
5 Observations:

After applying the image pre-processing techniques (deblurring, denoising, and contrast enhancement) on the original dataset, we observed the following effects on face detection performance using the Haar Cascade and MTCNN algorithms.

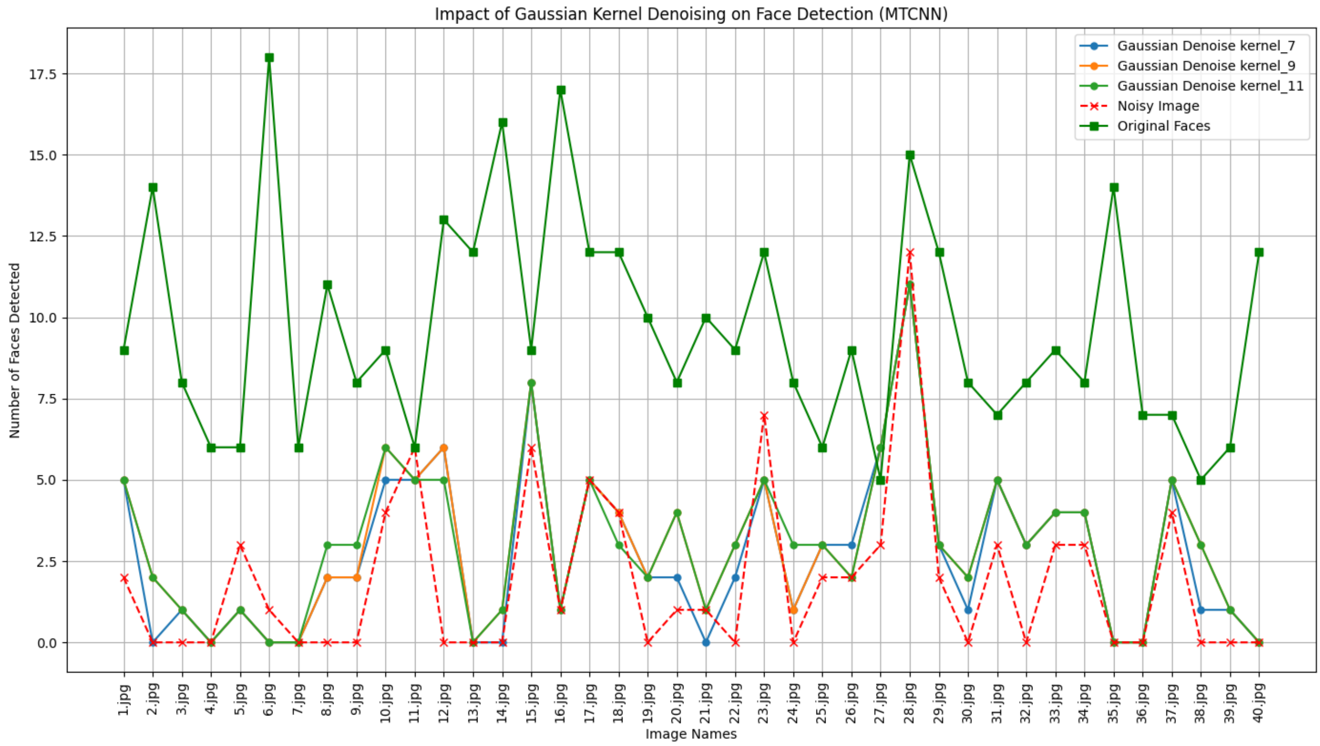
The plots for Haar Cascade :

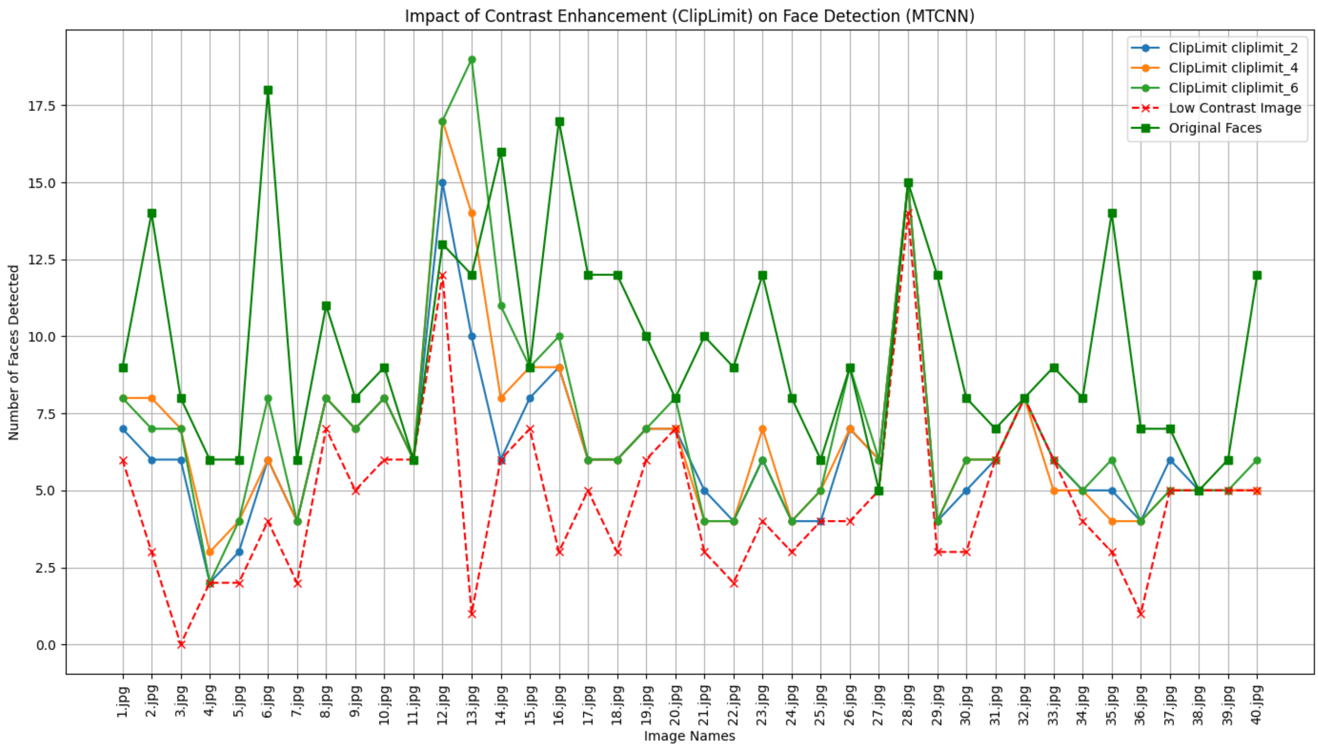






The plots of MTCNN::





5.1 Impact of Preprocessing Parameters

The effect of preprocessing on face detection was further analyzed by varying the parameters of the preprocessing techniques. This section highlights the observed impacts for deblurring, denoising, and contrast enhancement.

Deblurring For deblurring, we applied techniques such as Wiener filtering and Blind Deconvolution with varying parameters. The results indicate that deblurring had a minimal impact on improving face detection performance for both Haar Cascade and MTCNN:

- **Haar Cascade**: While not significantly improved, Haar Cascade was more responsive to deblurring compared to MTCNN. Some blurred images became detectable after applying the deblurring methods, particularly for smaller kernel sizes.

- **MTCNN**: The performance of MTCNN remained relatively stable, with negligible gains from deblurring. This behavior can be attributed to the algorithm's robustness and ability to work with degraded image features.

Denoising Denoising was analyzed using methods such as Gaussian smoothing and bilateral filtering, with parameters like kernel size and sigma being varied.

- **Haar Cascade**: The algorithm responded very positively to denoising, with face detection accuracy improving significantly for noisy images. The method effectively reduced noise that otherwise masked critical facial features, enabling Haar Cascade to detect faces that were initially missed.

- **MTCNN**: In contrast, MTCNN showed minimal improvement from denoising. Its advanced architecture, which leverages deep neural networks, already accounted for noise to a significant extent, resulting in only slight benefits from denoising methods.

Contrast Enhancement Contrast enhancement proved to be the most impactful preprocessing technique for both algorithms. Using methods such as CLAHE with varying clip limits and tile sizes, we observed marked improvements:

- **Haar Cascade**: The algorithm's face detection performance improved dramatically with contrast enhancement. Increased contrast made facial features more distinguishable, aligning with Haar Cascade's reliance on handcrafted feature detection.

- **MTCNN**: Contrast enhancement also benefited MTCNN significantly. While its baseline performance is already high, enhancing contrast improved the algorithm's ability to detect faces in challenging conditions such as low-light or low-contrast images.

6 Discussion and Conclusion

6.1 Effectiveness of Preprocessing Techniques

From the observations in our study, it is evident that preprocessing techniques significantly impact face detection performance for both Haar Cascade (Viola-Jones) and MTCNN algorithms. However, Haar Cascade demonstrates a notably higher sensitivity to preprocessing enhancements compared to MTCNN. Among the various techniques applied:

- **Contrast Enhancement** emerges as the most impactful preprocessing method for both algorithms. Haar Cascade showed substantial improvement in face detection accuracy, particularly with methods like CLAHE, where even minor enhancements in contrast resulted in the detection of previously undetected faces.

- MTCNN, while also responsive to contrast enhancement, exhibited comparatively smaller gains. This is because MTCNN inherently employs more robust feature extraction mechanisms that are less prone to degradation from contrast or noise.
- For other preprocessing methods like **denoising** and **deblurring**, Haar Cascade again showed pronounced benefits. In contrast, MTCNN demonstrated more stable performance, likely due to its deep learning architecture's ability to handle suboptimal image quality.

6.2 Which Face Detection Algorithm Benefits the Most?

When analyzing the benefits derived from preprocessing, it is evident that the Haar Cascade algorithm gains more from these techniques than MTCNN. This trend is apparent across all preprocessing methods, with the performance gap being especially pronounced in low-quality images where Haar Cascade would otherwise fail to detect faces.

This disparity can be attributed to the following reasons:

- **Algorithmic Characteristics**: Haar Cascade relies on handcrafted features, which are sensitive to variations in image quality, contrast, and noise. Preprocessing serves to bridge these gaps by enhancing the image's underlying features, making them more detectable by the algorithm. On the other hand, MTCNN uses deep learning techniques with robust feature extraction, allowing it to detect faces even in challenging conditions, albeit with less reliance on preprocessing.
- **Achieved Results**: The significant improvement seen with Haar Cascade highlights its dependency on preprocessing to optimize its performance. This aligns with the algorithm's simpler and more deterministic nature, where input quality has a direct impact on detection accuracy. MTCNN, while benefiting from preprocessing, already achieves higher baseline performance, meaning the relative gain from preprocessing is less pronounced.

In conclusion, preprocessing techniques like contrast enhancement, denoising, and deblurring play a crucial role in improving the accuracy of face detection algorithms. Traditional methods like Haar Cascade benefit the most, with preprocessing compensating for their inherent limitations. While MTCNN also shows improvement, its advanced architecture reduces its reliance on preprocessing, making it a more consistent performer across varied image qualities.