Face Detection and Deblurring with Laplacian Variance using HOG-based Dlib Detector and OpenCV

line 1: Ishika Mohan   
line 2: Btech 3 year AI&DS  
line 3: *NMIMS, navi mumbai*

line 1: Krisha patel   
line 2: Btech 3 year AI&DS  
line 3: *NMIMS, navi mumbai*

line 1: Garv Ubhan  
line 2: Btech 3 year AI&DS  
line 3: *NMIMS, navi mumbai*

*Abstract*— This paper presents an automated pipeline for face detection and quality assessment in images, integrating Histogram of Oriented Gradients (HOG) based face detection using Dlib and blurriness assessment via Laplacian variance. The proposed system detects faces, evaluates their sharpness, and applies deblurring techniques to blurry regions. The methodology also includes functionality for storing detected faces and visualizing sharpness levels through graphical plots. This approach has potential applications in quality assurance in photography, surveillance, and pre-processing for face recognition systems.

Keywords— Face Detection, Dlib, HOG, Laplacian Variance, Image Deblurring, OpenCV

# Introduction

Face detection and quality assessment are crucial in fields like facial recognition, image enhancement, and real-time surveillance, where image clarity significantly impacts downstream processing and recognition accuracy. Many real-world images suffer from motion blur or focus-related issues, complicating automatic recognition tasks. This paper introduces a system that combines the power of Dlib's Histogram of Oriented Gradients (HOG)-based face detection with a Laplacian-based blurriness metric to evaluate face quality. The approach involves detecting faces in a given image, identifying blurry regions, and optionally deblurring them to improve image clarity. This work aims to provide an effective, lightweight, and easily implementable solution suitable for various applications requiring real-time processing.

1. **2. Related Work**

Face detection is a well-researched area in computer vision, with solutions ranging from traditional approaches like HOG and Support Vector Machines (SVMs) to more recent deep learning-based architectures such as Convolutional Neural Networks (CNNs). Dlib's HOG-based face detector has proven to be an efficient and reliable method for face detection in static images and real-time applications [1]. Blurriness detection methods, particularly Laplacian variance, are commonly used for image quality assessment due to their simplicity and computational efficiency. Research by Michaeli and Irani [2] discusses the use of patch recurrence for deblurring, while Krishnan and Fergus [3] explore hyper-Laplacian priors for fast image deconvolution, highlighting the importance of edge preservation and sharpness in enhancing image quality.

1. **3. Methodology**

The proposed method involves five main steps: face detection, blurriness detection, deblurring, face cropping and storage, and annotated logging with graphical visualization. Each component of the system is described in detail below.

1. **3.1 Face Detection**

The face detection component employs Dlib's HOG-based detector, known for its robust performance and computational efficiency. HOG extracts features based on gradient orientations, making it effective for identifying human faces under various lighting conditions and poses. In the code, the detector is initialized with get\_frontal\_face\_detector(), and the detected faces are returned along with confidence scores.

**hog\_face\_detector = dlib.get\_frontal\_face\_detector()**

1. **3.2 Blurriness Detection**

After detecting faces, the system assesses the sharpness of each detected face region. This is achieved by converting the face region to grayscale and calculating the Laplacian variance, which serves as an indicator of image sharpness. A high Laplacian variance implies a sharp image, while a low variance suggests blurriness. The threshold for blurriness is set to 100.0, which serves as a heuristic for distinguishing clear faces from blurry ones.

**def is\_blurry(image, threshold=100.0): gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY) laplacian\_var = cv2.Laplacian(gray, cv2.CV\_64F).var() return laplacian\_var < threshold, laplacian\_var**

1. **3.3 Deblurring**

For faces identified as blurry, an optional deblurring process is applied. This process combines an edge-preserving filter and unsharp masking. The edge-preserving filter maintains important facial details while reducing noise, and the unsharp mask further sharpens the image by enhancing contrast along edges. This two-step approach results in improved clarity without introducing excessive artifacts.

**def deblur\_image(image): filtered = cv2.edgePreservingFilter(image, flags=1, sigma\_s=60, sigma\_r=0.4) blurred = cv2.GaussianBlur(filtered, (9, 9), 10.0) sharpened = cv2.addWeighted(filtered, 1.5, blurred, -0.5, 0) return sharpened**

1. **3.4 Face Cropping and Storage**

Detected face regions are cropped and saved to an output directory for easy access and further analysis. This step is essential for applications in surveillance or facial recognition, where detected faces need to be stored for later processing. Each face is saved with a unique identifier, enabling structured storage.

**Def save\_face\_crop(face\_region, face\_index, output\_dir="detected\_faces"): if not os.path.exists(output\_dir): os.makedirs(output\_dir) face\_path = os.path.join(output\_dir, f"face\_{face\_index}.jpeg") cv2.imwrite(face\_path, face\_region)**

1. **3.5 Image Processing and Logging**

The complete image processing pipeline involves face detection, blur classification, and annotation with rectangles around detected faces. For clear faces, a green box is drawn, while blurry faces are marked with a yellow box. Each face's Laplacian variance is logged for later analysis. Additionally, a graphical plot of each face's Laplacian variance is generated, providing a visual summary of face sharpness levels.

**def process\_image(image\_path, deblur\_faces=True, log\_path="image\_analysis\_log.txt"):**

**...**

**plt.bar(face\_labels, face\_laplacians, color='blue')**

**plt.axhline(y=100.0, color='red', linestyle='--', label="Blur Threshold")**

**plt.xlabel("Faces")**

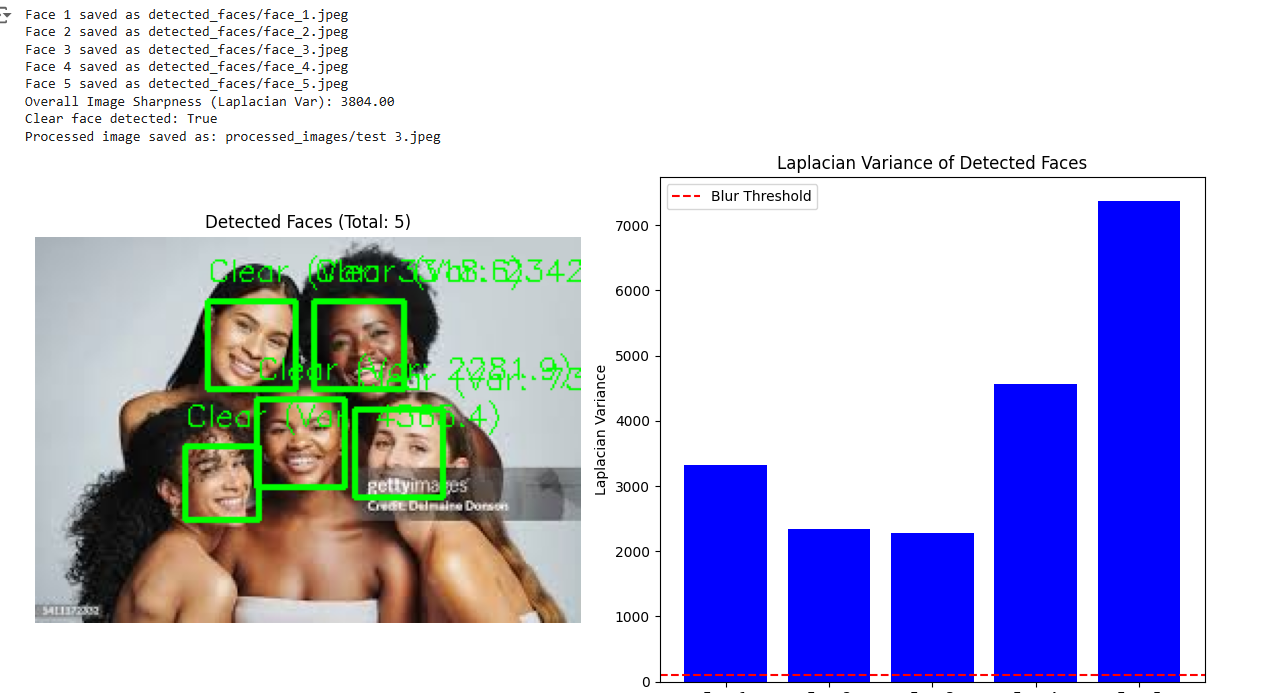
**plt.ylabel("Laplacian Variance")**

**plt.title("Laplacian Variance of Detected Faces")**

1. **4. Results and Discussion**

Experiments were conducted on images with varying qualities, including high-resolution images, low-light photographs, and images affected by motion blur. The system performed well, accurately detecting faces and classifying them as clear or blurry. In cases of blurry faces, deblurring noticeably enhanced the facial details, demonstrating the system's utility in enhancing low-quality images.

The Laplacian variance threshold of 100.0 was generally effective across various image resolutions, though it may need adjustment based on image source or resolution. The edge-preserving filter combined with unsharp masking improved clarity while minimizing artifacts, making the system suitable for real-time applications. The graphical plots of face sharpness levels allowed for quick visual evaluation, which can be valuable in quality assurance scenarios.



1. **5. Conclusion**

This paper presents an automated pipeline for face detection and quality assessment, utilizing HOG-based detection and Laplacian variance for sharpness evaluation. The system can identify blurry faces, enhance them, and provide visual feedback through graphical plots. The use of Dlib’s HOG detector with OpenCV's image processing capabilities offers a lightweight, efficient solution suitable for various applications requiring face quality assurance and pre-processing for facial recognition systems. Future work could include integrating more sophisticated deblurring algorithms, possibly through deep learning, to handle challenging blurring scenarios and improve face clarity further.

1. **References**
2. Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
3. Michaeli, T., & Irani, M. (2014). Blind Deblurring Using Internal Patch Recurrence. *European Conference on Computer Vision*.
4. Krishnan, D., & Fergus, R. (2009). Fast Image Deconvolution using Hyper-Laplacian Priors. *Advances in Neural Information Processing Systems*.

1. **Code File**

****