Investigating the Effect of Compression on Face Recognition with OpenCV

Mallellu Sai Prashanth[1(✉)], Ramesh Karnati[1], Muni Sekhar Velpuru[2], H. Venkateshwara Reddy[1], and Charmi R. Jani[3]

[1] Department of Computer Science and Engineering, Vardhaman College of Engineering, Hyderabad, India
saiprashanth08@ieee.org
[2] Department of Information Technology, Vardhaman College of Engineering, Hyderabad, India
[3] RK University, Rajkot, Gujarat, India

Abstract. In the recent years have seen an increase in interest in computer vision. Recognition is now one of the more effective and successful uses of image analysis and algorithms, and it has evolved from a niche to a popular area of computer vision research. The system will develop an application that would grant user access to a specific machine based on a thorough analysis of a person's facial features due to widespread curiosity and interest in the subject. Python and OpenCV will be used in the development of this application. by using a face detection algorithm, which locates and acknowledges faces in images but does not identify them. I also like to extract the feature in an image that displays each face. In certain pipelines, face alignment has been shown to improve face recognition accuracy. An application that tracks and recognizes faces in cameras and videos and has multiple uses. The project aims to investigate face detection with an open CV in depth. Since OpenCV is C-based, it can be run on any device that supports C. It performs admirably under Linux, Mac OS X, and Windows. Compared to OpenCV, MATLAB is far more expensive. MATLAB is approximately $2,150 while OpenCV is free. Due to its commercial, single-user license, even the basic version of Matlab is pricey. Furthermore, OpenCV is freely available due to its BSD license. Given that Java, which is derived from C, is the basis for MATLAB. As a result, when a script is run on MATLAB, the computer starts up by reading the code, translating it to Java, and then carrying out the script. Open CV, however, makes use of C/C++ library functions. Which aids in quicker execution by giving the computer the machine language code directly. When OpenCV is used, more time and resources are used for image processing and less for interpretation.

Keywords: Image · Face Detection · OpenCV · Computer Vision

1 Introduction

In the era of digital imagery, compression plays a vital role in reducing the storage requirements and transmission bandwidth of images. However, it is essential to understand the impact of compression on computer vision tasks to ensure reliable and accurate

X. Cheng (Ed): BROADNETS 2024, LNICST 602, pp. 79–88, 2025.
https://doi.org/10.1007/978-3-031-81171-5_8

results. One such critical task is face detection, which forms the basis for numerous applications like facial recognition, surveillance systems, and human-computer interaction. This research paper aims to investigate the impact of compression on utilizing the open-source computer vision library OpenCV for face detection widely employed in academia and industry. The study delves into how different compression algorithms affect the accuracy and robustness of face detection algorithms and explores potential trade-offs between image compression and face detection performance [1].

The compression process, typically employed to reduce the file size of images, inherently involves the loss of certain image details. These lost details may include facial features crucial for face detection algorithms, thereby affecting their ability to detect and localize human faces accurately. Understanding the extent to which compression affects face detection performance is of paramount importance for real-world applications that heavily rely on this technology. To conduct this investigation, we employ OpenCV's face detection algorithms, which encompass various methodologies such as Haar cascades and deep learning-based techniques. These algorithms serve as a benchmark to evaluate the impact of compression on face detection accuracy. The study focuses on widely used compression algorithms such as JPEG, which offer adjustable compression levels that trade off image quality for reduced file size [2].

The experimental methodology involves systematically compressing a diverse set of facial images at different compression levels and subsequently subjecting them to face detection algorithms using OpenCV. The performance of the face detection algorithms on the compressed images is compared against that of the original, uncompressed images. Various evaluation metrics such as detection accuracy, speed, and robustness are analyzed to quantify the impact of compression on face detection. The findings of this research are expected to shed light on the relationship between image compression and face detection performance. By identifying the compression levels at which face detection algorithms start to exhibit degradation in accuracy and reliability, this study can provide valuable insights for optimizing compression settings in real-world applications. Additionally, the research aims to provide guidelines for practitioners and developers who utilize face detection in scenarios where image compression is a necessary consideration [3].

In conclusion, this research paper presents an investigation into the impact of compression on face detection using OpenCV. Through a systematic evaluation of various compression algorithms and their effect on face detection performance, this study aims to contribute to the understanding of the trade-offs and considerations when employing image compression in face detection applications. The results and insights obtained from this research will pave the way for improved utilization of face detection algorithms in compressed image scenarios, enabling more efficient and reliable computer vision systems [4].

2 Related Work

Image compression algorithms, such as JPEG and JPEG2000, aim to reduce file sizes by exploiting redundancies in pixel data. JPEG, for instance, utilizes lossy compression that discards certain image details to achieve higher compression ratios. Familiarizing ourselves with different compression algorithms' characteristics is crucial to assess their impact on face detection [5].

Several studies have explored the influence of image compression on face detection accuracy. Research has shown that high compression ratios, particularly with lossy compression like JPEG, lead to increased false positives and reduced accuracy. It is important to understand how compression affects the performance of face detection algorithms [6].

Deep learning-based face detection algorithms have demonstrated remarkable accuracy and robustness. Research has examined the performance of these algorithms, such as SSD and YOLO, under different compression settings. The findings suggest that while deep learning-based methods generally outperform traditional approaches, they are still sensitive to high compression, resulting in degraded detection performance [6].

Transfer learning has emerged as a powerful technique in deep learning to address data scarcity. Some studies have explored the potential of training face detectors on compressed images and subsequently testing on uncompressed data. The research suggests that transfer learning from compressed images can mitigate the negative impact of compression on face detection performance. However, the effectiveness of this approach varies depending on the compression method and ratio [7].

To improve face detection accuracy on compressed images, researchers have proposed adaptation and enhancement techniques. Pre-processing methods, such as image enhancement and contrast normalization before applying face detection algorithms, have shown promising results. These techniques can help alleviate the impact of compression, although their effectiveness may depend on the specific compression method employed [8].

Understanding the impact of compression on face detection is crucial for real-world applications where images undergo compression during storage or transmission. Research has investigated the performance of face detection systems in surveillance scenarios where compressed video feeds are prevalent. The studies emphasize the need for robust face detection algorithms capable of handling the challenges posed by image compression in practical applications [9].

The literature survey reveals that image compression significantly affects the performance of face detection algorithms. Both traditional and deep learning-based approaches are impacted by compression, with deep learning methods showing better resilience to moderate compression levels. Transfer learning and adaptation techniques have shown promise in improving face detection accuracy on compressed images. However, further research is needed to address the complexities introduced by different compression methods and ratios [10].

3 Proposed Model

Architecture Diagram (Fig. 1):

The suggested setup is Using the MUHULANOBIC metric, faces existing in the complex background can be detected. Its foundation is the analysis of faces in two-dimensional photos of real-world scenes. It clearly utilizes the color segmentation procedure of the image being used as input. Thresholding the image in the hue color space is how the color is being segmented. This includes the effects of variations in the image's illumination on the uniqueness of the hue of human skin. After then, the image's results are combined for further analysis. The final step is to apply a restricted number of

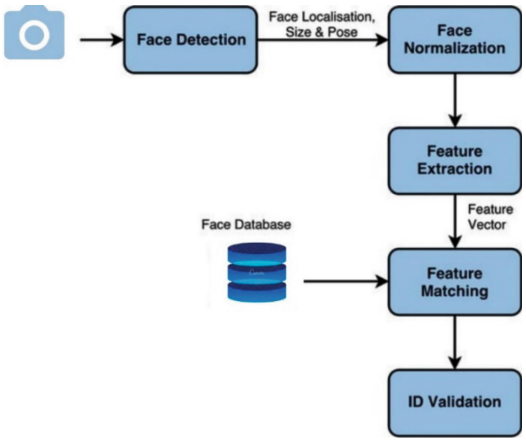

Fig. 1. Basic Architecture Diagram of proposed system

pixel accumulations to the median filter the resulting image. Ultimately, a multi-layer perceptron neural network is employed using the invariant moments as an input to distinguish between the faces and the remaining complicated backdrop. Three fundamental processes are involved in face recognition: face extraction, face recognition, and face detection. For any system to identify the position of the face, it must first encapsulate the image and then manage and record the essential elements. It records a number of variables, including skin tone and color, in order to identify the taken image. When an image from a camera or video is fed into face recognition software, the recognized image topic is produced as the output. Face cuts, structured and stylized angles, and different facial regions can all be considered facial features. Getting the features out of the camera is a face extraction process. Face detection involves removing the background and concentrating on the foreground, eliminating any other elements outside of the face region. Despite this, the system still has some limitations because it is unable to detect the head count, which could be caused by false positives for two faces with similar facial features or by overlapping faces. In a publication, Hussell Hardy suggests implementing face recognition using principal component analysis with four distance classifiers (Figs. 2 and 3).

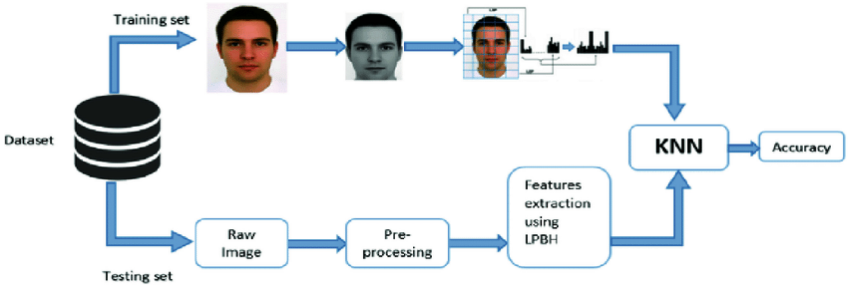

Fig. 2. Demonstrating Image JPEG Compression module

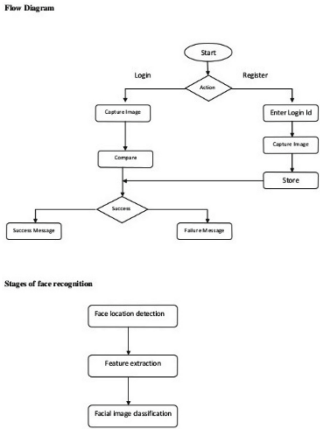

Fig. 3. Flow Chart Diagram of Image Detection using OpenCv

An image segementation, where users can perform for application, during registration process, we gather all the required information from users. To validate the users image, we are sending a verification token to the system, unless user verifies, he/she cannot login into web portal. After verification, users may login at any time. Portal consists of all the information like about our product, documentation and application to download.

Internal Images Concept

24 × 24 base window size, that would mean that this window would calculate over 180,000 features. Considering evaluating the difference in pixels for each feature. The Integral Image concept has been proposed as the solution for this computationally demanding process. Since the image is integral, all we need to know is the values of the four corners to find the total number of pixels under each rectangle. This means we do not need to add up each pixel by hand for us to calculate the sum of pixels in any feature window. All we have to do is use the four corner values to calculate the integral image. The following example will provide transparency for the process (Fig. 4).

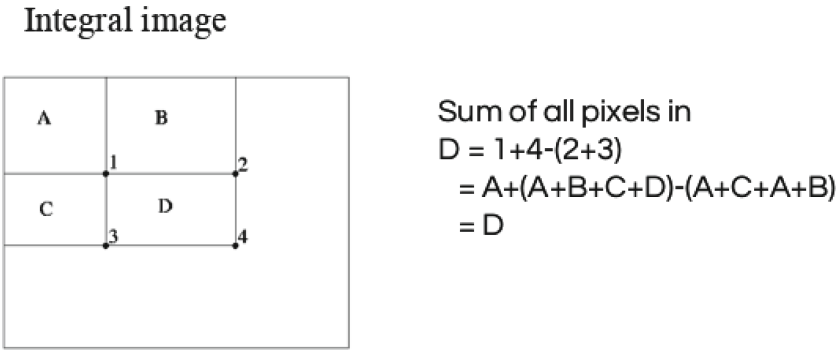

Fig. 4. Basic flow Diagram of Internal Image

We have developed application, compatible only for windows OS and implemented on different files varying from 1 mb to 30 mb of file size. Timing for image recognition as well as detection of such file are noted and compared with the respective recognition and detection timings of same files using Image Internals. As we are focusing only on security and not on timing as of now, we would like to work further on timing along with adding some more normalization techniques.

Explanation of Image Compression Functions
More than 13,000 face photos gathered from the internet are included in the Labelled Faces in the Wild Database set. Of these.jpg photos, 431 were arbitrarily chosen, and their average quality factor was 75. The majority of these pictures depict people's motivations during speeches or events. The images have varying resolutions, ranging from 233 × 409 to 410 × 450 pixels. There are multiple faces among the motives in this data set of images. This means that at high compression ratios, there's a good chance small faces won't be detected.

4 Results

See (Figs. 5, 6 and 7).

Fig. 5. No Image Detection for Preprocessing and No Normalization till 135 degress

Fig. 6. Multiple Image Detection

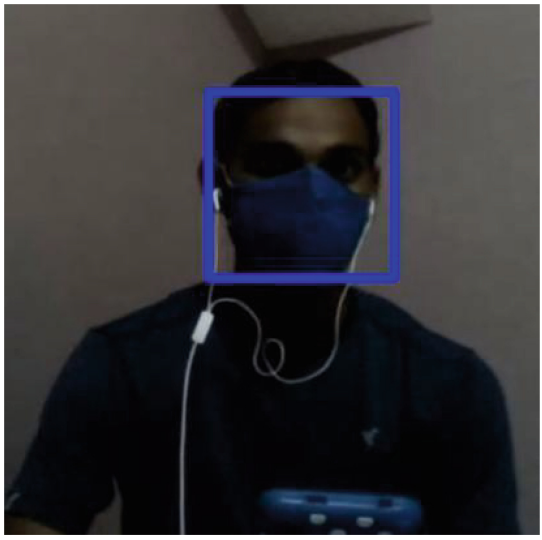

Fig. 7. Final Image Detection

4.1 Result Analysis

See (Figs. 8 and 9).

Existing Method	Proposed Method
Unknown faces cannot be classified with sufficient confidence	Unknown faces to the known classified with sufficient confidence
Compute 512-d face embeddings to quantify a face	Compute 128-d face embeddings to quantify a face
Automatic License Plate Recognition	Support Vector Machine for facial Recognization
Clustering, Representation and Classification	Clustering, Representation, Classification and Facial embedding
Facial landmarks	Facial landmarks with rotation and scaling
No Accuracy	Applied face alignment and cropping

Fig. 8. Comparision Analysis of Proposed Method & Existing Method

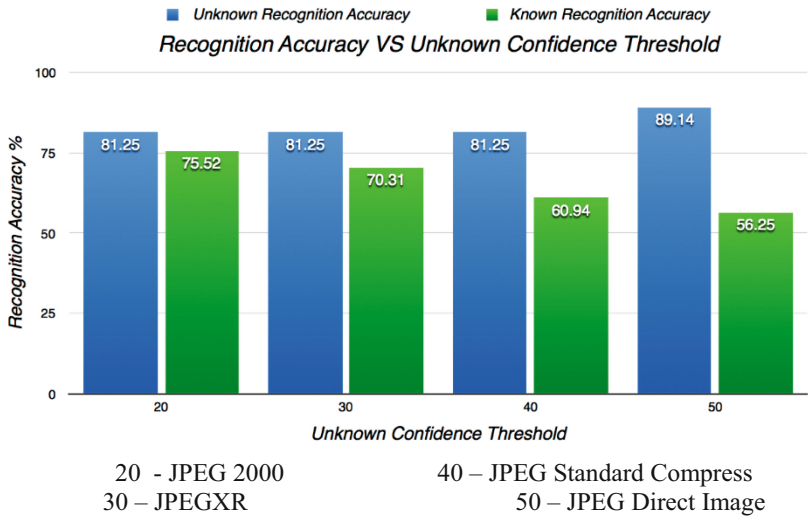

Fig. 9. Different Compression Standards

5 Conclusion

After studying different Image detection schemes, we get to know the pros and cons of each and possible attacks on each type. It turns out that, when comparing the similarity of the results obtained on pre-compressed and non-pre-compressed datasets, a small amount of JPEG precompression has no effect on the performance and ranking of the three compression methods. We have Made the Execution on the Open Cv Algorithm and produce the results at end. Prediction of the better Image Compression Performance using the Predicted values. Created a Python script to assist in collecting labeled faces. In the

context of face detection using sets of Haar-like features, our experiment shows that JPEG XR and JPEG2000 outperform JPEG, particularly when the images are compressed at high ratios. JPEG XR is a good substitute for JPEG2000 in all circumstances because of the noticeably lower computational demand and marginally better face detection results. We present ± one standard deviation along with the mean values, confirming the first observation, in order to highlight the distinctions between JPEG XR and JPEG 2000. If we look at the relationship between PSNR and detection accuracy, we find that JPEG is correctly predicted by its PSNR values to be the worst algorithm; on the other hand, the small benefits of JPEG XR over JPEG 2000 are not correctly predicted by PSNR. Furthermore, when comparing the similarity of the results obtained on pre-compressed and non-pre-compressed datasets, it has been found that a small amount of JPEG precompression has no effect on the performance and ranking of the three compression algorithms.

Future Scope

The development of face detection technology is only getting started; there are a plethora of further uses for this technology. The different concepts are:It might be used in taxis operated by nighttime drivers. An algorithm can be set up so that if the driver blinks their eyes for longer than five seconds, the car's alarm system may sound, alerting everyone to the driver's drowsiness. Although it might not be able to completely eliminate the accident rates caused by this, it might be able to lower them. While adults who find themselves lost can return home, what about children who are lost, abducted to work as slaves, or even worse, where there are fears that the young females are being exploited as prostitution, Hundreds of lives could be spared from torture by producing a severe version of these faces and applying face identification algorithms. Regardless of the station—airport or metro—it can be utilized there. We may potentially lower the crime rate if face detection technology were installed at the entryway to every station and compared with the worldwide list of wanted individuals from every police agency.

References

1. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, pp. 07–49. University of Massachusetts, Amherst (2007)
2. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I-511–I-518 (2001)
3. Wallace, G.K.: The JPEG still picture compression standard. IEEE Trans. Consum. Electron. **38**(1), 18–34 (1992)
4. Rakshit, S., Monro, D.M.: An evaluation of image sampling and compression for human iris recognition. IEEE Trans. Inf. Forensics Secur. **2**(3), 605–612 (2007)
5. Zhuang, S.-S., Lai, S.-H.: Face detection directly from h.264 compressed video with convolutional neural network. In: Proceedings of the IEEE International Conference on Image Processing, ICIP 2009, pp. 2485–2488 (2009)
6. Hämmerle-Uhl, J., Karnutsch, M., Uhl, A.: Evolutionary optimisation of JPEG2000 part 2 wavelet packet structures for polar iris image compression. In: Ruiz-Shulcloper, J., Sanniti di Baja, G. (eds.) CIARP 2013, Part I. LNCS, vol. 8258, pp. 391–398. Springer, Heidelberg (2013)

7. Goyal, K., Agarwal, K., Kumar, R.: Face detection and tracking: Using OpenCV. In: 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 474-478 (2017). https://doi.org/10.1109/ICECA.2017.8203730
8. Saini, N., Kaur, S., Singh, H.: A review: face detection methods and algorithms. Int. J. Eng. Res. Technol. **2**(6) (2013). www.ijert.org
9. Kirby, M., Sirovich, L.: Application of the Karhunen-Loeve procedure for the characterization of human faces. IEEE Trans. Pattern Anal. Machine Intell. **12**(1), 103–108 (1990)
10. Chen, D.-S., Liu, Z.-K.: Generalized haar-like features for fast face detection. In: Conference on Machine Learning and Cybernetics, 2007. Hong Kong, pp. 2131–2135 (2011)