INTERNATIONAL SCIENTIFIC AND VOCATIONAL JOURNAL (ISVOS JOURNAL)

Vol.: 8 Issue: 2 Date: 31.12.2024 Received: 24.06.2024 Accepted: 27.08.2024 Final Version: 27.12.2024

ISVOS Journal, 2024, 8(2): 103-122 – DOI: 10.47897/bilmes.1501078

Facial Tracking, Recognition, and Utilizing Gaussian Blur In Face Recognition Sytems Via The OpenCv Library

Muhammed Emin Necipsoy [a,1], Atilla Ergüzen [b]

[a] General Directorate of Health for Borders and Coasts of Türkiye, İstanbul, Türkiye
ORCID ID: 0000-0003-3978-6365
[b] Computer Engineering, Kırıkkale University, Kırıklale, Türkiye
ORCID ID: 0000-0003-4562-2578

Abstract

Face recognition technology attracts great attention in many technological areas. The development of face recognition algorithms has made significant contributions to the elimination of deficiencies in the field of image processing. Especially image processing libraries such as OpenCV provide a reliable and regularly updated platform for researchers and developers. OpenCv, which includes face recognition algorithms, is an image processing library that facilitates image processing. Some people may not want their faces to be seen in videos, movies or live broadcasts, and objectionable images and harmful products such as cigarettes and alcohol may need to be censored. In this case, the Gaussian filter comes to our rescue. The Gaussian filter is a filter widely used in image processing techniques and known for its blurring feature. The Gaussian filter is also called blurring in image processing software. The Python language is a programming language that can work independently of the platform. The Python language contains many libraries and is easy to program. The OpenCv library, like many other libraries, has generally been used with the Python language because it works very well with the Python language and is easily programmed. Many projects developed with Python language and OpenCv can be seen in academic sources. The aim of this study is to perform face recognition using OpenCV library and automatically apply Gaussian filter to recognized faces. All existing software does not automatically blur the desired faces. Doing this process manually is both time-consuming and jeopardizes the protection of privacy due to the unnoticed parts of the manual application process. Possible users of this project include televisions, production companies, broadcasters and YouTubers. This project can contribute to more effective protection of privacy and save time. This article can provide a method for researchers, industry experts and academics.

Keywords: “Face recognition, OpenCv, Gaussian filter, image processing, bluring.”

1. Introduction

With the development of technology, facial recognition algorithms, which are an important component in various applications such as personal identification, security enhancement, automation and human-machine interaction, have also made great progress [1-7]. Face detection and recognition algorithms contribute to overcoming many difficulties in daily life, protecting privacy, automation systems and various fields by accurately identifying individuals and verifying their identities with the help of learning-based algorithms such as artificial intelligence and machine learning through the analysis of facial features.

In recent years, significant advances have been made in facial recognition technology thanks to innovations in software and hardware that increase accuracy and efficiency. In particular, the adaptation of artificial intelligence, deep learning algorithms and big data analytics has increased the speed and reliability of facial recognition systems [5-46]. Image processing libraries, especially the OpenCv image processing library, provide facilities and libraries for easy implementation of face recognition algorithms [47]. This process of advancement in the field of facial detection and recognition has not only improved security applications but also expanded the scope of facial recognition in various fields [12].

Image processing and computer vision are rapidly growing fields that are essential for many industrial and academic applications [40]. OpenCv (Open Source Computer Vision Library) is an open source library that is widely used in these areas. OpenCv, first developed by Intel in 2000 and constantly updated, offers a wide range of tools for researchers and engineers [44]. The library's availability in a variety of programming languages such as C++, Python, Java and MATLAB increases its versatility and makes it invaluable for the development of complex image processing applications.

[1] Corresponding Author
E-mail Address: muhammedemin.tombak@saglik.gov.tr

OpenCv's flexibility makes it a preferred choice for a variety of applications, from research to commercial products. It provides tools for basic image processing, object detection, motion analysis, face recognition, 3D modeling and deep learning adaptations [44]. This powerful feature set allows developers to efficiently implement advanced algorithms to quickly prototype and deploy their projects in image processing.

In this article, a software project in which OpenCV is used to detect, learn and recognize faces, and then the Gaussian filter, in other words the blur filter, available in OpenCV, is automatically applied to the selected learned face. In the software project that is the subject of this academic article, the machine learning method and the Haar Cascade algorithm were preferred for face recognition. A common smoothing and blurring method, the Gaussian filter is used to blur images and reduce noise. This method effectively protects people's privacy by making unwanted facial features unrecognizable, whereby the Gaussian filter is widely used, especially in facial recognition and privacy protection applications.

Innovation is encouraged in a variety of disciplines, including the development and application of image processing techniques, computer vision, medical imaging, remote sensing, security systems and content production. Image filters are critical in the field of image processing. Image filters improve image quality, remove features, add new features, or highlight certain areas. In this context, the Gaussian filter is used effectively to blur unwanted parts of an image.

Facial recognition technology has become an integral part of daily life, and as its scope expands and this technology develops, the accuracy and efficiency of the techniques and methods of facial detection and recognition algorithms are becoming increasingly better [7]. Continuous improvements in these technologies are necessary to meet increasing security and privacy demands in a variety of applications, from personal devices to large-scale surveillance systems.

This study aims to demonstrate the potential of the Gaussian filter in facial recognition systems and the developments being made in this field, and to provide valuable information to researchers and industry experts. Automatically hiding faces in videos is very important for privacy and security. Additionally, this process saves cost and time. For example, automatic detection and hiding of markers in videos or inappropriate content in edited footage can significantly reduce manual effort. This process is becoming increasingly important in light of increasing data protection concerns and legal regulations.

With the development of technology, facial recognition algorithms, which are an important component in various applications such as personal identification, security enhancement, automation and human-machine interaction, have also made great progress [7]. Face detection and recognition algorithms contribute to overcoming many difficulties in daily life, protecting privacy, automation systems and various fields by accurately identifying individuals and verifying their identities with the help of learning-based algorithms such as artificial intelligence and machine learning through the analysis of facial features.

In recent years, significant advances have been made in facial recognition technology thanks to innovations in software and hardware that increase accuracy and efficiency. In particular, the adaptation of artificial intelligence, deep learning algorithms and big data analytics has increased the speed and reliability of facial recognition systems [5]. Image processing libraries, especially the OpenCv image processing library, provide facilities and libraries for easy implementation of face recognition algorithms [47]. This process of advancement in the field of facial detection and recognition has not only improved security applications but also expanded the scope of facial recognition in various fields [12].

Image processing and computer vision are rapidly growing fields that are essential for many industrial and academic applications. OpenCv (Open Source Computer Vision Library) is an open source library that is widely used in these areas. OpenCv, first developed by Intel in 2000 and constantly updated, offers a wide range of tools for researchers and engineers [44]. The library's availability in a variety of programming languages such as C++, Python, Java and MATLAB increases its versatility and makes it invaluable for the development of complex image processing applications.

OpenCv's flexibility makes it a preferred choice for a variety of applications, from research to commercial products. It provides tools for basic image processing, object detection, motion analysis, face recognition, 3D modeling and deep learning adaptations [44]. This powerful feature set allows developers to efficiently implement advanced algorithms to quickly prototype and deploy their projects in image processing.

In this article, a software project in which OpenCV is used to detect, learn and recognize faces, and then the Gaussian filter, in other words the blur filter, available in OpenCV, is automatically applied to the selected learned face. In the software project that is the subject of this academic article, the machine learning method and the Haar Cascade algorithm were preferred for face recognition. A common smoothing and blurring method, the Gaussian filter is used to blur images and reduce noise. This method effectively protects people's privacy by making unwanted facial features unrecognizable, whereby the Gaussian filter is widely used, especially in facial recognition and privacy protection applications.

Innovation is encouraged in a variety of disciplines, including the development and application of image processing techniques, computer vision, medical imaging, remote sensing, security systems and content production. Image filters are critical in the field of image processing. Image filters improve image quality, remove features, add new features, or highlight certain areas. In this context, the Gaussian filter is used effectively to blur unwanted parts of an image.

Facial recognition technology has become an integral part of daily life, and as its scope expands and this technology develops, the accuracy and efficiency of the techniques and methods of facial detection and recognition algorithms are becoming increasingly better [7]. Continuous improvements in these technologies are necessary to meet increasing security and privacy demands in a variety of applications, from personal devices to large-scale surveillance systems.

This study aims to demonstrate the potential of the Gaussian filter in facial recognition systems and the developments being made in this field, and to provide valuable information to researchers and industry experts. Automatically hiding faces in videos is very important for privacy and security. Additionally, this process saves cost and time. For example, automatic detection and hiding of markers in videos or inappropriate content in edited footage can significantly reduce manual effort. This process is becoming increasingly important in light of increasing data protection concerns and legal regulations.

2. Literature Review

Facial recognition technology is used to identify or verify individuals by creating mathematical models of faces in images and comparing these models to a database or training set [4]. This section provides a literature review of the use of OpenCV and the Gaussian filter in face recognition and describes the different principles and image filters used in this process.

Face recognition technology has evolved considerably with the development of various algorithms and methods. These methods can be broadly divided into two main types: spatial filters and frequency filters. Spatial filters directly manipulate the individual pixels of an image and perform tasks such as smoothing, sharpening or edge detection to improve image quality. Frequency filters, on the other hand, analyze the frequency components of the image and offer an alternative approach to improving image characteristics by taking the frequency range into account. Both spatial and frequency filters are important to improve the accuracy of face recognition [5].

By using these techniques, the process of face recognition can be refined and improved to ensure better performance and reliability. This comprehensive review of the literature emphasizes the critical role of image processing tools such as OpenCV and the application of Gaussian filters in the advancement of face recognition technology.

2.1. Some of The Most Commonly Used Algorithms for Face Recognition

2.1.1. Eigenfaces

Eigenfaces is a method primarily used to assess the variance of faces and to encode and decode facial features using machine learning techniques. This approach uses a collection of "Eigenfaces"," which are essentially basic face components identified by applying principal component analysis (PCA) to an extensive database of faces. The Eigenfaces algorithm has been recognized for its remarkable accuracy in face recognition tasks [6]. However, it is sensitive to illumination variations, that can significantly degrade its performance under different lighting conditions. Despite its robustness, this method struggles when exposed to inconsistent lighting conditions, which is a critical aspect in practical applications.

2.1.2. Fisherfaces

Similar to the Eigenfaces method, the Fisherfaces method uses the Fisher criterion to improve the discrimination between different face classes. This technique was developed to identify the features that most effectively discriminate between different face classes and thus improve the accuracy of face recognition processes. A major advantage of the Fisherfaces method is its lower sensitivity to illumination variations and facial expressions [28]. Consequently, this method proves to be more reliable across a range of illumination conditions and facial expressions, making it very effective in practical applications.

The Fisherfaces approach, which focuses on maximizing class separability, overcomes some of the limitations found in other methods, that can falter under inconsistent lighting conditions or changing facial expressions. This robustness is particularly valuable in real-world scenarios where lighting conditions can be unpredictable and facial expressions vary frequently. Therefore, the Fisherfaces method is characterized by its resilience and improved performance in different and dynamic environments.

2.1.3. Local Binary Patterns Histograms (LBPH)

The LBPH method uses a two-stage process for face recognition. First, the face image is converted into local binary patterns, which then serve as the basis for calculating the corresponding histograms. This method is characterized by its simplicity and speed as well as its ability to obtain texture information [19]. Therefore, the LBPH method is often used in real-time applications thanks to its minimal computational cost and high speed.

Moreover, the process starts with the conversion of the face image into local binary patterns. This step is crucial as it effectively captures the texture features of the image, which are essential for accurate recognition. The histograms of these patterns are then calculated, which form the core of the LBPH method.

What makes LBPH particularly attractive is its efficiency. The algorithm is designed to be both simple and fast, which is a significant advantage in practice. Furthermore, the preservation of texture information ensures that the essential features of the face are preserved throughout the recognition process [19]. This is particularly important in applications where speed and accuracy are of paramount importance.

Due to its advantages, the LBPH method is often used in scenarios that require real-time face recognition. Due to its low computational cost, it can also be used in environments with limited computing power without sacrificing performance. This makes it a versatile tool in various fields, from security systems to user authentication procedures.

In the first phase, the conversion of the face image into local binary patterns, the image is analyzed at a granular level to extract detailed texture information. This information is then summarized into histograms that provide a comprehensive representation of the features of the face. These histograms are used in the subsequent recognition phase to compare and identify faces with high accuracy.

The speed and efficiency of LBPH is underlined by its ability to process and recognize faces quickly, making it suitable for real-time applications. The design of the method ensures that it remains computationally lightweight, enabling fast processing even in hardware-constrained environments. This balance of speed, simplicity and accuracy makes LBPH a preferred choice in many practical applications [19].

To summarize, the two-step process of the LBPH method - —converting images into local binary patterns and computing histograms — enables efficient and accurate face recognition. The advantages of simplicity, speed and texture preservation make it ideal for real-time applications, especially when computational resources are limited. This method is characterized by its ability to deliver high performance with minimal computational effort, which underpins its widespread use in various fields.

2.1.4. Facenet

FaceNet is an advanced face recognition method based on the principles of deep learning. The core of this technique is the embedding of face images in a high-dimensional vector space, which enables the measurement of similarity between faces using the Euclidean distance metric. This embedding process is central to the effectiveness of FaceNet as it ensures that faces are represented in a way that enables accurate recognition.

One of the most notable advantages of FaceNet is its high accuracy and scalability. When trained on large-scale datasets, FaceNet shows exceptional performance, making it a preferred choice for applications that require high precision [25]. This high-dimensional embedding allows FaceNet to capture intricate details of face images, enhancing its recognition capabilities.

The process begins with embedding face images into a high-dimensional vector space. This step is critical as it converts the images into vectors that can be easily compared using Euclidean distance. The similarity between two face images is then determined by calculating the distance between their corresponding vectors, with smaller distances indicating greater similarity.

The effectiveness of FaceNet is largely due to the foundation of deep learning. By utilizing large data sets for training, FaceNet can achieve remarkable accuracy. This makes it particularly useful for applications where precision is important, such as security systems and identity verification processes [25]. Its scalability further increases its usefulness, as it can process large amounts of data without sacrificing performance.

Another important aspect of the method is the use of the Euclidean distance metric to measure similarity. This metric provides a simple yet powerful means of comparing face vectors and ensures that similar faces are accurately recognized. The use of high-dimensional vectors ensures that even subtle differences between faces are captured, contributing to the high accuracy of the method.

FaceNet's deep learning approach and the embedding of high-dimensional vectors make it possible to achieve exceptional performance in face recognition. Its ability to scale with large datasets and maintain high accuracy makes it a preferred solution in many demanding applications [25]. The combination of these features emphasizes the robustness and reliability of FaceNet in various face recognition scenarios.

In summary, FaceNet is characterized by its innovative use of deep learning and vector space embedding. These elements work together to create a method that is both highly accurate and scalable. By converting face images into high-dimensional vectors and using the Euclidean distance metric to measure similarity, FaceNet ensures accurate recognition even under challenging conditions. The success of this method in processing large data sets and maintaining performance highlights its potential for wide application in various fields that require reliable face recognition technology.

2.1.5. Deep Learning and Artificial Neural Networks

Deep learning has made a huge impact in the field of facial recognition. In particular, Convolutional Neural Networks (CNNs) have achieved great success in facial recognition tasks. [39] CNNs effectively recognize facial features by processing and learning an image through various layers. Commonly used CNN models such as VGGFace, ResNet, and Inception are known for their high accuracy rates and strong generalization abilities [3].

VGGFace is a deep learning-based facial recognition model and is based on VGG16 architecture. This model achieves high accuracy rates by training on large data sets. ResNet, on the other hand, makes it possible to train deeper networks by using skip connections to increase the learning speed between layers. Inception, on the other hand, provides a more effective feature mapping by combining filters of different sizes in the same layer. These models demonstrate superior performance in face recognition tasks by using the power of deep learning [3].

2.1.6. Histogram of Oriented Gradients (HOG)

HOG is a method used in face recognition tasks by analyzing the edge structures and gradient directions of an image. HOG features allow facial recognition algorithms to capture structural information of the face. This method is widely used especially in face detection and verification processes and offers high performance [38].

HOG calculates gradient directions and magnitudes for each pixel, then uses this information to create cellular histograms. These histograms are then normalized to obtain more stable and scalable representations of facial features. The HOG method is very popular in face recognition and detection applications as it effectively captures the structural features of the face [38].

2.1.7. Support Vector Machines (SVM)

Support Vector Machines (SVM) is a powerful machine learning algorithm used in classification and regression analysis. In facial recognition systems, SVM performs facial recognition tasks by parsing facial features in a high-dimensional space. Especially when combined with feature extraction methods such as HOG, SVM can improve face recognition accuracy [6].

SVM is used to identify hyperplanes that maximize the separation between classes. This method performs accurate classification by parsing data points in the feature space. Especially in face recognition tasks, SVM can achieve effective results with input data such as HOG features [6].

2.1.8. DeepFace

DeepFace is a facial recognition system developed by Facebook and based on deep learning methods. This method learns and recognizes facial images using deep neural networks. DeepFace creates 3D models of faces, providing high accuracy regardless of different angles and lighting conditions. [37]

The DeepFace method uses 3D modeling of the face, analyzes each part of the face separately and combines these analyzes to perform the face recognition task. This method shows outstanding performance and delivers high accuracy rates when trained on large datasets. [37]

2.1.9. Scale-Invariant Feature Transform (SIFT)

Scale-Invariant Feature Transform (SIFT) is an algorithm used to recognize objects in images independently of scale and transformation changes. In facial recognition systems, it determines the similarities between faces by extracting the distinctive features of the face. SIFT is known for being particularly robust to scale and transformation changes [36].

The SIFT method detects points of interest in the image and performs the face recognition task by analyzing the local features of these points. This method provides high accuracy even at different scales and transformations and is widely used in face recognition systems [36].

The selection of face recognition algorithms requires careful consideration of various factors, as each algorithm offers different advantages and disadvantages depending on the data set, performance criteria and application scenarios. Therefore, the selection of the most suitable algorithm is a crucial element in the research process.

Different algorithms perform particularly well under certain conditions. Some algorithms may perform better on large data sets and offer high accuracy and robustness. Others, on the other hand, may be optimized for speed and computational efficiency and are therefore better suited to real-time applications where fast processing is essential. The selection process therefore cannot be generalized, but must be tailored to the specific requirements of each application.

Performance criteria are another critical factor. In certain applications, accuracy is more important than anything else, e.g. in security systems where false positives or negatives can have significant consequences. In other scenarios, such as user authentication on personal devices, speed and resource efficiency may be more important. The strengths and weaknesses of each algorithm in terms of accuracy, speed and computational effort must be thoroughly evaluated to find the best solution for the intended use [25].

The application scenarios also influence the choice of algorithm. For example, algorithms that work well in controlled environments with consistent lighting and positioning may not be as effective under dynamic, real-world conditions. On the other hand, some algorithms are designed to handle different lighting, angles and even occlusions, making them more versatile for different environments. To select an algorithm that performs reliably, it is important to understand the specific challenges and requirements of the application context.

The process of selecting a face recognition algorithm is therefore multi-layered and must take into account the specific requirements of the dataset, performance criteria and application scenarios. This comprehensive evaluation ensures that the chosen algorithm matches the research objectives and practical requirements of the application [25]. In this way, researchers can utilize the strengths of different algorithms to achieve optimal results in their face recognition projects.

2.2. Image and Video Filters

Various filters have been developed to meet specific requirements for images and videos. These filters modify the image or video to meet specific requirements. For example, old family photos could be in black and white. Certain filters can colorize these black and white images. Another example is the need to make a person's face unrecognizable in a video to protect their privacy.

In the project discussed here, the Gaussian filter was used to protect privacy by blurring. The Gaussian filter is a widely used technique in image processing, especially for smoothing images and reducing noise. It recalculates the value of each pixel based on a weighted average of the surrounding pixels, with the weights decreasing according to the Gaussian function as the distance from the center pixel increases [25]. This process softens sharp edges and effectively reduces noise.

The application of the Gaussian filter is not limited to two-dimensional images, but can also be applied to three-dimensional data, demonstrating its flexibility and versatility. It is particularly useful in facial recognition and privacy applications, where it can blur unwanted areas to ensure the confidentiality of personal data. This filter can focus on specific areas of an image to remove distractions and highlight the important parts [33].

In video recordings, the Gaussian filter can make certain faces unrecognizable if necessary. For example, to preserve privacy, it can be used to make certain faces unrecognizable so that they are difficult to identify. This aspect is crucial in scenarios where the anonymity of individuals must be preserved.

To summarize, it can be said that different filters serve different purposes in image and video processing. The Gaussian filter is characterized by its effectiveness in blurring and noise reduction, making it a valuable tool for privacy protection and image enhancement. Its ability to be applied to both 2D and 3D data further enhances its usefulness in various applications.

2.3. OpenCv

OpenCV, or the Open Source Computer Vision Library, is a highly regarded open source library in the field of image processing and computer vision. Originally developed by Intel in 2000, OpenCV provides a wealth of tools that are useful for both researchers and engineers [44].

This platform-independent library is available in several programming languages, including C++, Python, Java and MATLAB. This level of flexibility ensures that OpenCV is widely used in various applications, from academic research to commercial products [41].

The OpenCV library has an extensive range of tools, from basic image processing functions to advanced applications such as object recognition and motion analysis. It is best known for its role in the development and implementation of face recognition algorithms. OpenCV excels at recognising faces primarily through the use of Haar cascade classifiers, which are known for their high accuracy and efficiency in performing face recognition tasks [47]. Haar cascade classifiers provide fast and effective solutions and are therefore ideal for real-time applications.

One of the key strengths of OpenCV is its comprehensive set of functionalities that cover both basic and advanced requirements in image processing and computer vision. This versatility, coupled with support for multiple programming languages, has cemented OpenCV's status as one of the most important tools in the field. Whether used in research labs or integrated into commercial products, OpenCV's robust capabilities continue to drive innovation and efficiency.

In summary, OpenCV is an indispensable resource in the field of image processing and computer vision, providing indispensable tools and unparalleled flexibility. Its success in face recognition, particularly with Haar cascade classifiers, underscores its value for both real-time and high-accuracy applications. Thus, OpenCV remains a cornerstone in the toolbox of researchers and engineers alike, enabling a multitude of technological advances.

2.4. Other Image and Video Filters

In addition to the Gaussian filter, there are many other filters used in image and video processing. These filters use a variety of methods to meet specific requirements and improve image quality. Here are some commonly used filters:

2.4.1. Median Filter

The median filter is widely used to reduce noise in image processing tasks. This filter works by taking the median of the pixels in its local neighborhood for each pixel of the image. The median filter has the ability to effectively reduce sharp noises, especially salt and pepper noise [3].

The median filter uses the median of neighboring pixels to determine the value of each pixel in an image, which helps reduce sharp noise. This method is especially effective for reducing noise while maintaining image quality.

2.4.2. Sobel Filter

Sobel filter is a filter frequently used in edge detection processes. This filter calculates gradients in vertical and horizontal directions to detect edges in the image. The Sobel filter is used to determine the orientation and sharpness of edges and thus helps determine object boundaries [34].

The Sobel filter uses gradient calculations to highlight edges in the image, which is useful in object recognition and segmentation. Edge highlighting is critical to understanding the structural properties of the image.

2.4.3. Laplace Filter

The Laplace filter is used to identify sharp edges by calculating second-order derivatives in the image. This filter makes edges stand out and detects areas of rapid change in the image. Laplace filter is widely used in edge detection and sharpening [6].

The Laplace filter uses second-order derivatives to detect sudden brightness changes in the image. This is especially useful for highlighting edges and highlighting image details.

2.4.4. Bilateral Filter

A bilateral filter is a filter used to reduce noise while preserving the edges in the image. This filter takes weighted averages of pixels based on their spatial proximity and color similarity. The bilateral filter is especially effective for smoothing operations by preserving edges [33].

The Binary filter smoothes pixels by taking into account both spatial and color information. This method is especially preferred in image enhancement applications, thanks to its ability to reduce noise while preserving edges.

2.4.5. Wiener Filter

Wiener filter is a statistical filter used to reduce noise and blur in the image. This filter applies an optimal smoothing for each pixel by calculating the local variance of an image. The Wiener filter is used specifically to reduce blur and noise simultaneously [32].

The Wiener filter applies optimal smoothing by analyzing the local features of an image, thereby reducing both noise and blur. This method is effective in image enhancement and restoration processes.

3. Method and Dataset

OpenCV, a highly customizable image processing library, is compatible with numerous programming languages. This study focuses on the development of an automatic face detection and blurring application using Python and the OpenCV library. Python was chosen for its straightforward image processing capabilities and platform independence.

In the application created with OpenCV, the Haar Cascade method was used for face recognition. This technique, originally developed by Alfred Haar, facilitates the detection of objects and faces in images and can even recognize facial expressions.

The Haar cascade method includes two basic approaches to facial feature extraction: the geometric feature principle and the aspect principle. These principles significantly improve the accuracy and reliability of face recognition under different conditions.

The versatility of OpenCV also extends to its ability to handle complex image processing tasks with relative ease. Its robust functionality combined with Python's user-friendly syntax makes it an ideal choice for developing sophisticated applications. By using these tools, researchers can achieve a high level of precision in face recognition, which is crucial for numerous applications, from security systems to user interface enhancement.

The integration of OpenCV with Python is an example of a strong synergy in the field of image processing, providing researchers and developers with a powerful toolkit to advance their projects. The effectiveness of the Haar Cascade method in face recognition underlines the potential of these technologies to deliver reliable and efficient solutions in real-world scenarios.

3.1. Geometric Property Principle

The human face consists of several components, such as eyes, nose, mouth, beard and mustache. Although these features can vary in size and shape from person to person, their position on the face is usually relatively uniform. This uniformity is due to the arrangement of the facial components according to a certain ratio, commonly referred to as the golden ratio.

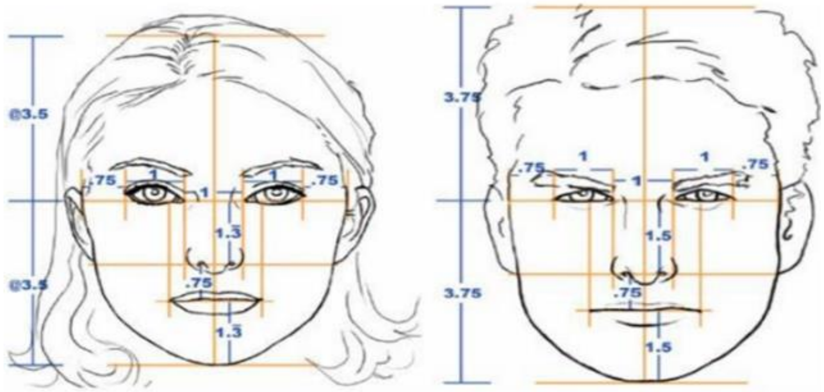

Fig. 1. Environment-Adjusted Kuznets Curve [51].

In the field of face recognition, the first phase of identification consists of extracting a feature vector that represents the face geometry based on the structure and positioning of its components. In this method, the main focus is on pupil detection. Once the pupils are identified, other facial features are then determined. Typically, the arrangement of these facial features is the same for about 95% of individuals.

The process begins with the identification of the pupils, which serve as important reference points. The positions of the other facial components are then assigned accordingly. This systematic approach ensures a high degree of accuracy in face recognition and utilizes the natural symmetry and proportions inherent in the human face.

In addition, understanding the geometric relationship between facial features helps to improving the reliability of face recognition systems. By using advanced algorithms and utilizing the golden ratio, these systems can achieve remarkable accuracy in identifying and verifying individuals.

To summarize, the uniform arrangement of facial features based on the golden ratio forms the basis for effective face recognition techniques. The careful process of recognizing key features such as the pupils and then matching other components ensures that recognition systems can operate with high accuracy and reliability in various applications.

3.2. Aspectual Principle

A pattern vector is created by applying visual filters and haar samples embellished with visual principles, analogous to Gabor filters, to specific regions of a face or to the entire face. The following section summarizes the information about the haar samples used in this study.

Haar samples, some of which were shown in Figure 2, are digital visual elements used in object recognition and diagnosis. During the recognition process, primary features of haar tufts, such as lines, corners and central points, are usually used. The process of identifying facial components in an image requires a comprehensive scan of the entire image.

In figure 2, a selection of Haar patterns are shown.

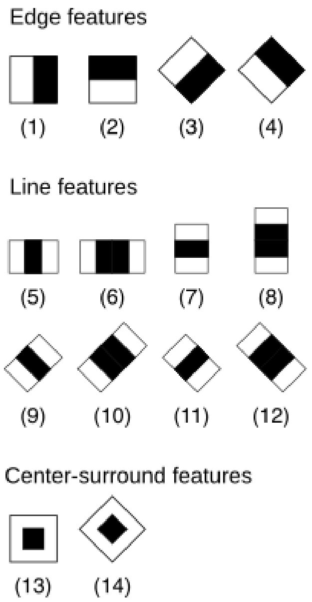

Fig. 2. Some Haar Samples.

In order to recognize human images of different sizes, it is essential to classify the classifier in an appropriate way. An effective solution is to use Haar filters instead of resizing the image [51]. In figure 3, the images of the Haar patterns used for recognizing eyes in a face are shown.

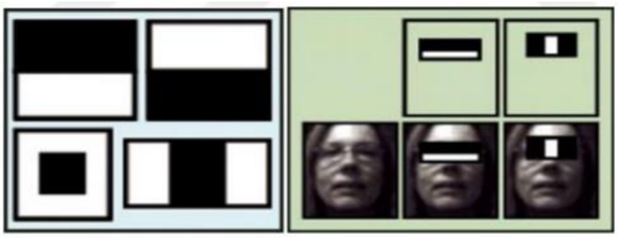

Fig. 3. Haar samples and Eye Detection [51]

In the context of face recognition, the mean value of the dark pixel area is subtracted from the mean value of the light pixel area in the image. If the resulting value exceeds a predefined threshold set during the learning phase, the Haar feature is considered applicable to the image [45].

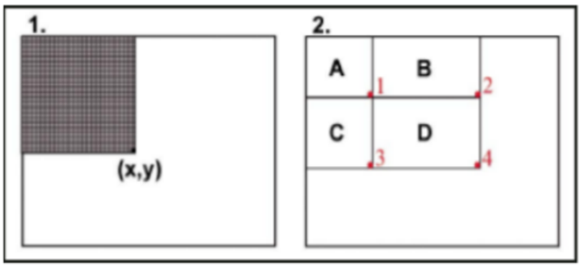

Fig. 4. The Process of Integrating an Image [52].

An integral image method is used to identify images in different quadrants and detect the presence or absence of thousands or even millions of Haar features in each image segment. The term "integral" refers to the linking of small units together. Figure 4 illustrates the integration process of an image. Jones and Viola used the AdaBoost method of machine learning to apply their Haar features in determining the optimal threshold.

Some notable Haar features are the edge feature, the line feature, and the center-periphery feature (quadrilaterality). These features enable the detection of different facial components, such as mouth, nose and ears, in one image. In addition, this method can recognize various other objects. As shown in Figure 3, the eyes were successfully identified using the Haar cascade method.

Before implementing the Haar cascade method, it is important to train the objects to be recognized. These objects can range from trees to license plates to logos. The cascade file in XML format is created as a result of the training process. In this phase, positive images that contain the object in question are compared with negative images that do not contain the object and thus subjected to a comprehensive training process. The cascade file can be created using this method, but there are also ready-made files for object recognition.

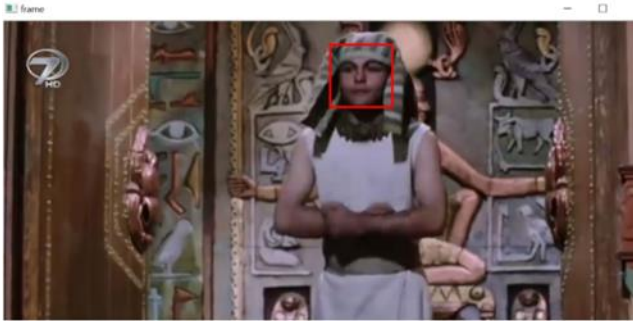

Fig. 5. Face of A Random Person Who was not Trained in The Project [42].

In figure 5, it is illustrated that the ability to recognize the face of any person in an image [42]. In figure 5, it is shown that the project places a red square around the face of a person who is not known in its dataset. In the study, the system was presented with images of specific people that it attempted to identify. As can be seen in Figure 6, the project successfully identified all the faces in the image and was able to recognize the face of a trained person [42]. In Figure 6, it can be seen that the project puts a red square around the face of a person who is not known in its dataset and a blue square around the face of the person who is known in its dataset. The system has drawn a blue square around the face of the known person, labeled it with the person's name (Yuzarseph) and enclosed the face of the unrecognized person in a red square [42].

Fig. 6. The Recognized Face of The Person Who has been Trained in The project and The Unrecognized Face of A Random Person Who has not been Trained in The Project in Question [42].

This process is similar to the way the human brain distinguishes faces in received images and recognizes familiar people. From an early age, the human brain learns general object names, characteristics and sizes. This learned data can be compared with a cascade file. The brain then identifies and distinguishes familiar people among the general objects. For example, when a baby opens its eyes and observes the world, it learns to distinguish faces, particularly those of its mother, father, siblings and relatives.

3.3. Image Filters

In addition to face recognition, the OpenCV library offers a variety of filters for image processing. These filters are widely used on platforms ranging from computers to cell phones and facilitate various image processing tasks. Image processing is a field in that digital images are analyzed and modified using various techniques. Libraries such as OpenCV offer numerous filter functions to help with these tasks, ranging from basic to advanced.

First, the pyrUp function enlarges an image and typically applies a Gaussian filter to estimate new pixels, doubling the image size in both dimensions [2]. Conversely, the pyrDown function downsizes an image, also using a Gaussian filter for resampling, effectively halving the image size [7].

Another important function is pyrMeanShiftFiltering, an image segmentation technique that combines color space and geometric space. This filter merges regions with similar colors and reduces noise while preserving edges, improving image segmentation by taking color and spatial information into account [16].

The boxFilter function smoothes an image by averaging the pixels within a certain kernel size. This blurs the image to reduce noise, but it can also attenuate edge details [14]. For more general filtering purposes, the filter2D function applies a two-dimensional kernel to an image and performs a convolution operation that can be customized for edge detection, blurring, or sharpening [29].

The Scharr filter, a derivative filter used to detect edges in an image, offers optimized results compared to the Sobel filter by providing better precision and less noise, making it ideal for edge detection tasks [3]. In addition, sepFilter2D applies a two-dimensional filter by splitting it into discrete components. This splits a 2D kernel into two 1D kernels, which reduces computational costs, especially for large kernels [2].

When building an image pyramid, the buildPyramid function displays images at different resolution levels, a technique useful for analyzing and blending images with multiple scales [1]. The GaussianBlur filter, known for its effectiveness in smoothing and noise reduction, uses a Gaussian function to blur an image, creating a natural blur that minimally affects edges [16].

For edge detection and image sharpening, the Laplacian filter uses the second derivative to highlight regions with rapid intensity changes and emphasize edges and details in the image [29]. Finally, the Sobel filter uses a series of kernels to detect edges in both horizontal and vertical directions, which is often used to emphasize edges in an image [3].

The OpenCV library's comprehensive set of filters and image processing functions, such as those mentioned above, provide researchers and engineers with robust tools that facilitate various applications in computer vision and image analysis.

3.4. The Gauss Filter

In addition to face detection, a Gaussian blur filter was used in this project to make the detected faces unrecognizable. The Gaussian blur filter is used to make objects in an image unrecognizable that should actually remain unseen. In films, for example, it is used to censor cigarettes, brands and alcohol or to make certain faces in images and videos unrecognizable.

The Gaussian filter, which is widely used in image processing, serves as a spatial filter to blur images and reduce noise. It was named after the German mathematician Carl Friedrich Gauss and is based on the principles of the Gaussian distribution [43]. This filter works by recalculating each pixel value in an image as a weighted average of its neighboring pixels. The weights used in these recalculations are determined by a Gaussian function that decreases with increasing distance from the center pixel [16].

The central mathematical basis of the Gaussian filter is the two-dimensional Gaussian function. This function assigns a weight to each pixel in the image, whereby the weights are distributed symmetrically around the central pixel. The general formula for the Gaussian function is showed below:

$$G(x,y) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2+y^2}{2\pi\sigma^2}\right) \tag{1}$$

The standard deviation of the Gaussian function which is represented in this paper by the number 1, which is denoted by σ, determines the filter size, while x and y represent the pixel coordinates [20].

The Gaussian filter smoothes high-frequency components in the image (e.g. sharp edges and noise) while emphasizing low-frequency components (e.g. broad structural areas). This leads to a reduction of details in the image, while the broad structures are retained. The ability of the Gaussian filter to preserve edges gives it an advantage over other averaging filters [8]. It is often used in image processing applications to protect privacy by blurring faces in images and to reduce noise in medical images [21]. In addition, the Gaussian filter can serve as a preprocessing step in face recognition systems to smooth facial images and thus improve the performance of recognition algorithms [21].

One of the main advantages of the Gaussian filter is its simplicity and effectiveness. It can be applied to both two-dimensional (2D) and three-dimensional (3D) data, which emphasizes its flexibility and versatility [50]. Despite its edge-preserving properties, the Gaussian filter can lead to edge blurring for very large σ-values. Therefore, the selection of a suitable σ-value depending on the application is crucial [7].

Overall, the Gaussian filter is an indispensable tool in various image processing applications as it effectively smoothes images while preserving important structural details.

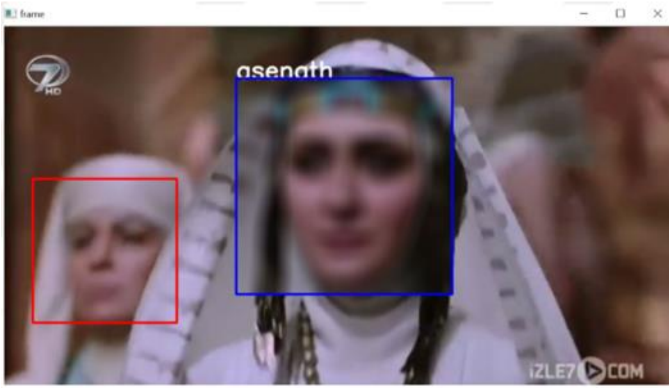

Fig. 7. Gaussian Blur Applied Face of The Person Who is Trained by The Project and Face of A Random Person Who is not Trained by The Project in question [42].

3.5. Project Introduction

As shown in Figure 7, this study demonstrates the automatic application of the Gaussian blur to a recognized face by the software. The intensity of the Gaussian blur filter can be adjusted as needed, providing flexibility depending on the requirements of the application [42].

In this study, OpenCV and Python were used to develop software that can recognize faces and apply the Gaussian blur to those faces. The software was carefully designed for the automatic application of blurring in multimedia content to ensure privacy protection. The main motivation for this project was the lack of such automatic blurring features in existing software, which highlighted the need for a tool that could seamlessly integrate privacy protection measures.

The development utilized the capabilities of OpenCV, a powerful open source library for computer vision tasks, and Python, which is known for its simplicity and effectiveness in handling image processing tasks. By combining these tools, the goal was to create a solution that not only identifies faces in images and videos, but also effectively applies Gaussian blurring to obscure them when necessary.

In essence, the project fills a significant gap in current multimedia software by providing an automated method for improving privacy. The adjustable intensity of the Gaussian blur filter further enhances the utility of the software and makes it suitable for various scenarios where privacy is of utmost importance.

3.6. Working Principle

The project was developed using the robust image processing capabilities of OpenCV combined with the versatility of Python. The workflow involves several key steps.

First, the software recognizes faces using OpenCV's pre-trained Haar cascade classifiers. These classifiers, which have been trained with various positive and negative images, are able to recognize unique facial features. Once the recognition phase is completed, the application proceeds to the next step.

In the second step, a Gaussian blur filter is applied to the detected faces so that they are no longer recognizable. This ensures anonymity and privacy protection in the processed images. The Gaussian blur recalculates each pixel value based on a weighted average of its neighbors, effectively blurring the image.

Face recognition, an important aspect of this project, is based on Haar cascade classifiers. These classifiers have been carefully trained with different image sets so that they can accurately identify specific facial features. After the detection phase, the Gaussian blur filter is applied, a standard technique in image processing to reduce details and noise while preserving the overall structure.

To summarize, this project successfully integrates face detection and blurring functions with OpenCV and Python. The key steps include detecting faces with Haar classifiers and then applying a Gaussian blur to these faces. The approach not only ensures privacy protection, but also demonstrates the powerful possibilities of combining these technologies in image processing tasks.

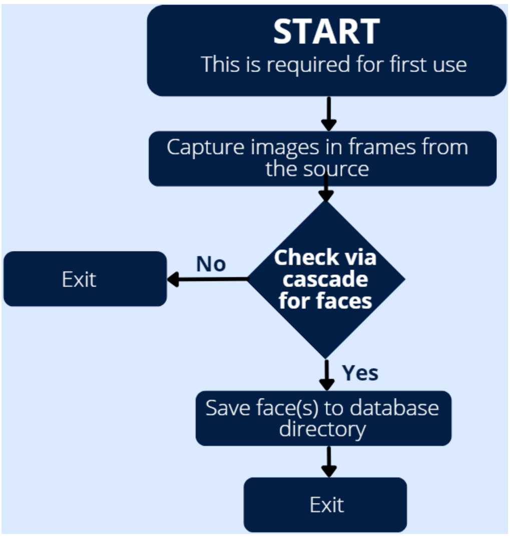

Fig. 8 (a). The following flow diagram illustrates the structure of the project's face capture module.

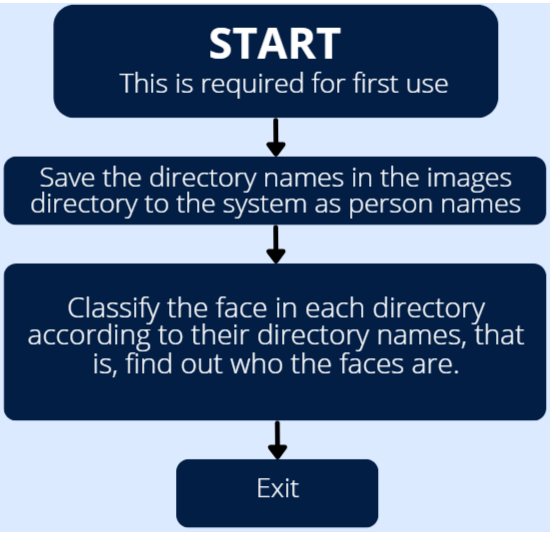

Fig. 8 (b). The following flow diagram illustrates the structure of the project's learning module.

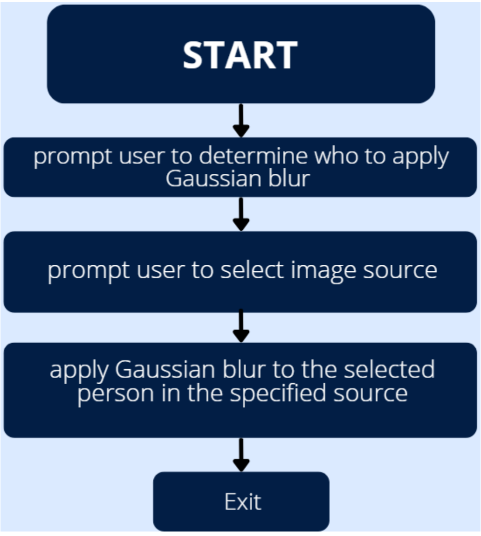

Fig. 8 (c). The following flow diagram illustrates the structure of the project's face recognition and blur module.

3.7. Working Method and Algorithm

Figure 8 provides a visual representation of the flowchart of the project. In the following sections, the development process and the software algorithm are explained in detail.

The development process started with identifying the core requirements for face recognition and blurring functionalities. Initially, OpenCV's powerful libraries were selected due to their robust image processing capabilities. Python was chosen due to its flexibility and extensive support for OpenCV.

After the initial setup, the software algorithm was developed in a series of systematic steps. First, face recognition is performed using pre-trained Haar cascade classifiers provided by OpenCV. These classifiers have been trained with a set of positive and negative images and are able to accurately recognize facial features.

Once the faces are detected, the next step is to apply a Gaussian blur filter to these recognized faces. This filter recalculates the pixel values based on the weighted average of the neighboring pixels, making the faces unrecognizable. The intensity of the Gaussian blur can be adjusted as required to ensure optimal privacy protection.

Technically, the face recognition phase relies heavily on Haar classifiers. These classifiers are designed to recognize certain features by processing the image in several stages, each increasing in complexity and specificity. The robustness of this method lies in its ability to deal with variations in lighting, angles and facial expressions.

After recognition, the Gaussian blur filter is applied. This filter, named after the German mathematician Carl Friedrich Gauss, uses a Gaussian function to weight the pixel values. The mathematical basis of this filter ensures that high-frequency components, such as sharp edges and noise, are smoothed out, while low-frequency components, such as wider structures, are retained.

In summary, the project's development process and software algorithm have been carefully designed to ensure efficient detection and blurring of faces. By utilizing the power of OpenCV and Python, the software achieves high accuracy in face detection and effectively applies the Gaussian blur filter to protect privacy.

3.7.1. Installation and Requirements

A Python environment first had to be set up for the project. Python was chosen due to its extensive library support and seamless integration options. Key libraries included OpenCV, which is known for its robust image processing capabilities, and NumPy, which is indispensable for numerical operations.

3.7.2. Face Detection

First, Haar cascade classifiers are loaded. These classifiers were trained with a large number of positive (face images) and negative (non-face images) examples. In each image of the video stream, the system searches for facial features. In this search, Haar features in different image regions are analyzed and a cascade model is used to classify and recognize faces.

3.7.3. Applying Gaussian Blur

Once faces are detected, the Gaussian blur filter is applied to the detected areas. The Gaussian blur calculates the weighted average of the pixel values surrounding a particular pixel, thus making the face unrecognizable and ensuring privacy. During application, each detected face region is blurred with a specific sigma value, obscuring the facial details and making the faces unrecognizable.

Figure 9 shows the face of a person who is not registered in the system. In figure 9, it is seen that the project puts a red square around the face of a person who is not known in its dataset

In Figure 10, we see the face of a person who is known to the system, but to whom no blurring has been applied.

As can be seen in Figure 11, the face of a registered person is outlined in blue, while the face of a non-registered person is outlined in red. The software is able to recognize multiple faces in a single visual frame.

Figure 12 illustrates the project's image directory. Faces to be trained to the software are stored in directories labeled with their respective names. The software's learning module is then executed, which completes the learning process by creating a cascade file based on this data.

In Figure 13, the software automatically blurs faces that cannot be recognized by the cascade file due to their low resolution, thus ensuring privacy [42]. The software has a minimum resolution threshold; if a detected face falls below this threshold, it is automatically blurred. In figure 13, a face is marked with the green frame that is blurred due to its low resolution.

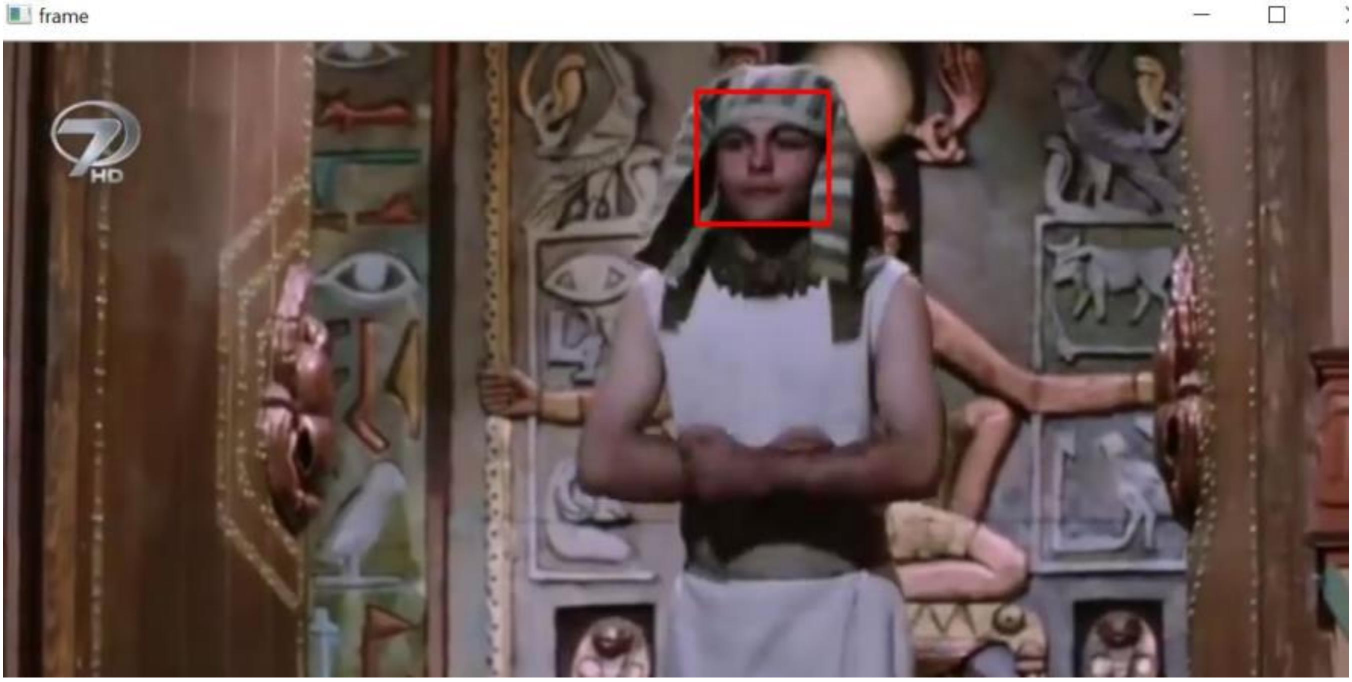

Fig. 9: An Unrecognized Face [42].

Fig. 10: The Face That Gaussian Blur has not been Applied of A Person Who is Trained by The Project [42].

Fig. 11: Face of Trained Person in The Software and Untrained Person [42].

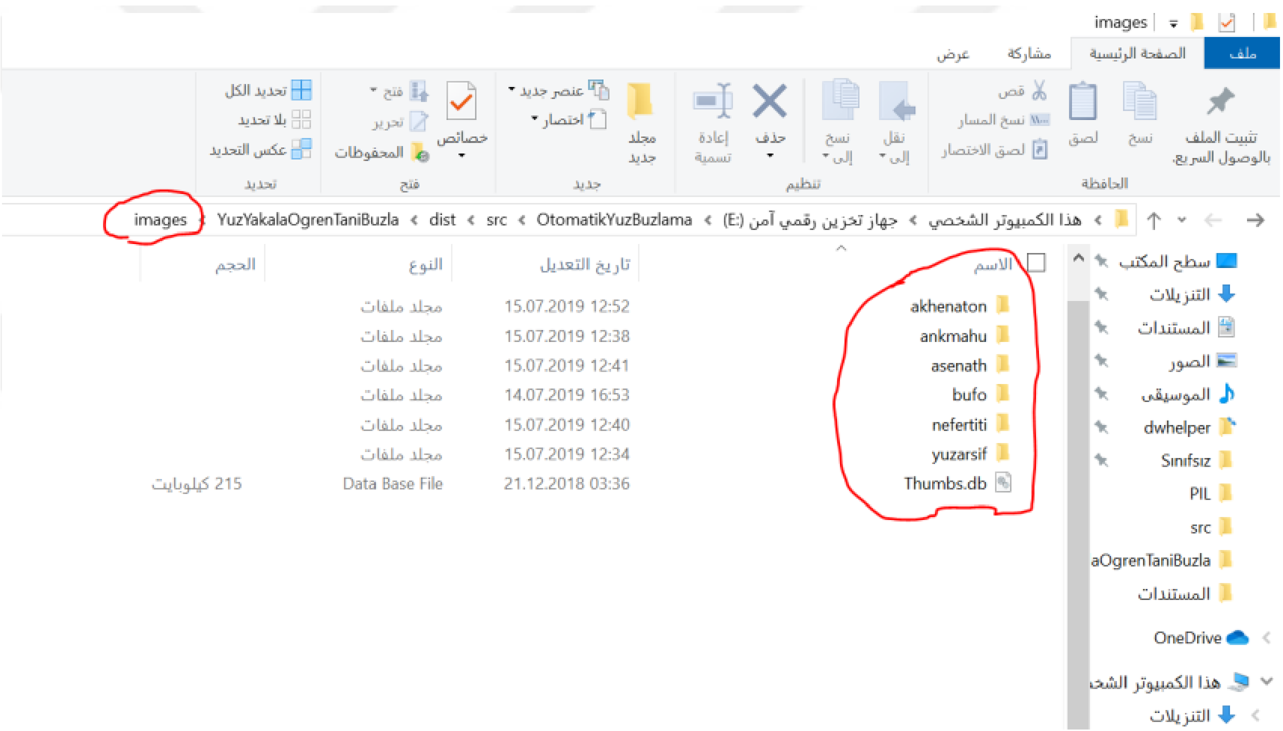

Fig. 12: Images Directory

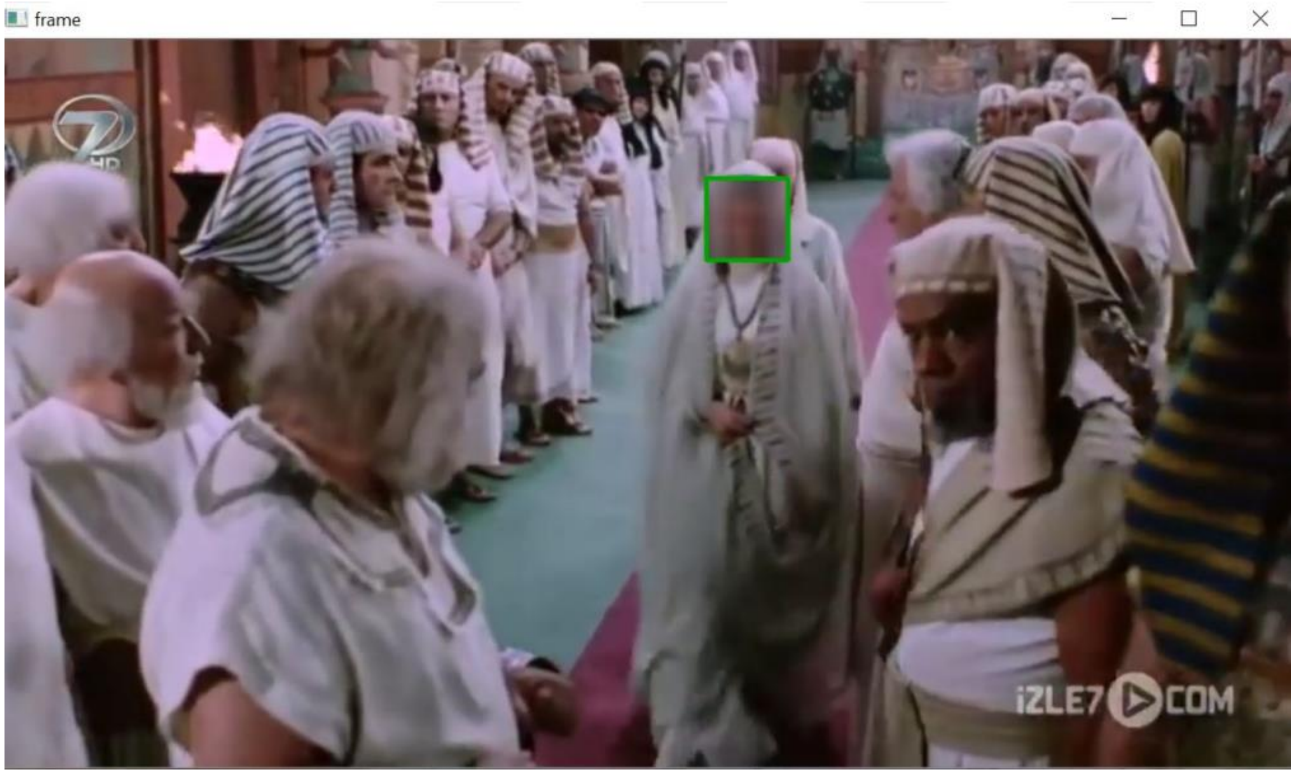

Fig. 13: Application of Gaussian Blur to Low-Resolution Faces [42]

4. Results

OpenCV is a widely used library for image processing on various platforms. Python is a user-friendly and platform-independent language that makes audio and video processing effortless [2]. In this study, software was developed that performs face recognition with OpenCV using Python and optionally applies Gaussian blur to the detected faces. The need for this software arises from the lack of existing software that performs the blurring process automatically [28]. For images with multiple objects that need to be blurred, each object must be blurred individually [44].

While the blurring process is relatively simple for still images, it is much more difficult for videos [43]. This project aims to solve this problem and was tested on a segment of the series that tells the well-known story of Prophet Joseph (PBUH), also known as Yuzarseph [42].

Figures 5, 6 and 7 summarize these tests. The success of the project is satisfactory, although several factors diminish this success. These factors include:

- Changes in lighting conditions: These have a significant impact on the accuracy of face recognition.
- The limitation of OpenCV to a single cascade file of a particular type: Processing multiple cascade files within a single file is challenging for OpenCV [27].
- Degradation of image clarity and resolution: Image resolution is crucial for the quality of face recognition and blurring.
- Degradation of image clarity and resolution: Image resolution is critical to the quality of face recognition and blurring.
- The inability to capture all angles of the subject in a single cascade file: The limited coverage of perspective in a single cascade file reduces the robustness of the recognition process [7].

To solve these problems, the following solutions were found:

- Consideration of illumination conditions: Ensuring uniform and appropriate illumination during image acquisition can improve recognition accuracy.

- Ensuring high image resolution: Using high-resolution images improves the quality of face recognition and subsequent blurring [3].

- Creating different threads and assigning different cascades to these threads: To overcome the limitation of OpenCV in terms of the number of cascades that can be stored in a single file, different threads can be created, each assigned to different cascades [34].

- Inclusion of comprehensive visual representations when creating cascade files: Including multiple viewpoints and perspectives of the face in the cascade file improves the robustness of recognition.

This approach ensures that the developed software effectively addresses the identified challenges, thus improving the overall efficiency and reliability of face recognition and blurring process in video sequences. Future work could focus on the integration of machine learning techniques to further improve the recognition accuracy under different conditions [1].

References

[1] C. M. Bishop, Pattern recognition and machine learning. Springer, 2006.

[2] G. Bradski, and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc, 2008.

[3] R. C. Gonzalez, and R. E. Woods, Digital image processing. Prentice Hall, 2008.

[4] R. C. Gonzalez, and R. E. Woods, Digital image processing (4th ed.). Pearson, 2018.

[5] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[6] A. Jain, Fundamentals of digital image processing. Prentice-Hall. Press, 1989.

[7] A. Jain and S. Li, Handbook of Face Recognition. Springer, 2011.

[8] A. K. Jain, R. Kasturi and B. G. Schunck, Machine vision. McGraw-Hill, 1995.

[9] V. M. Patel and R. Chellappa, Sparse representations and compressive sensing for imaging and vision. Springer, 2014.

[10] S. J. D. Prince, Computer vision: Models, learning, and inference. Cambridge University Press, 2012.

[11] J. C. Russ, The image processing handbook. CRC press, 2011.

[12] Datagen. Facial recognition algorithms and libraries you should know, 2022. [Online]. Available: https://www.datagen.tech/blog/facial-recognition-algorithms-and-libraries-you-should-know.

[13] R. Szeliski, Computer vision: Algorithms and applications. Springer, 2010.

[14] S. Kulkarni, Understanding image filtering techniques in image processing, 2023.. [Online]. Available: https://www.imageprovision.com/articles/understanding-image-filtering-techniques-in-image-processing.

[15] T. Ahonen, A. Hadid, and M. Pietikäinen, Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, 2006. [Abstract]. Available: ProQuest, https://www.scirp.org/reference/referencespapers?referenceid=2037884.

[16] P. J. Burt and E. H. Adelson, The Laplacian Pyramid as a Compact Image Code. IEEE Transactions on Communications, vol. 31, no. 4, pp. 532-540, 1983. https://doi.org/10.1109/TCOM.1983.1095851.

[17] J. Azmeen, and D. J. Borah, (2021). Face recognition techniques and challenges: A review. In N. Marriwala, C. C. Tripathi, S. Jain, & S. Mathapathi (Eds.), Soft computing for intelligent systems (pp. 345-360). Springer, 2021. https://doi.org/10.1007/978-981-16-1048-6_27

[18] A. Bhat, R. K. Jha, and V. Kedia, Robust face detection and recognition using image processing and OpenCV. In 2022 6th International Conference on Computing Methodologies and Communication (ICCMC). IEEE. , 2022. https://doi.org/10.1109/ICCMC53470.2022.9753792

[19] V. Bhavani, K. S. Priya, A. K. Sirivarshitha, and K. Sravani, An approach for face detection and face recognition using OpenCV and face recognition libraries in Python. In 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE., 2023. https://doi.org/10.1109/ICACCS57279.2023.10113066

[20] L. M. Gladence, M. M. Khan, S. Mohammad, et al. 2022 6th International Conference on Computing Methodologies and Communication (ICCMC). IEEE., 2022. https://doi.org/10.1109/ICCMC53470.2022

[21] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, Attribute and Simile Classifiers for Face Verification. Proceedings of the IEEE International Conference on Computer Vision, 2009.

[22] R. Lienhart, and J. Maydt, An Extended Set of Haar-like Features for Rapid Object Detection. Proceedings of the IEEE International Conference on Image Processing. IEEE, 2002.

[23] C. Liu, and H. Wechsler, Enhanced Fisher Linear Discriminant Models for Face Recognition. Proceedings of the IEEE International Conference on Image Processing. IEEE, 2003.

[24] Masi, I., Tran, A. T., Hassner, T., and Medioni, G. Pose-aware face recognition in the wild. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2018.

[25] O. M. Parkhi, A. Vedaldi and A. Zisserman, Deep face recognition. British Machine Vision Conference, 2015.

[26] S. Ren, K. He, R.Girshick and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems, 2015.

[27] P. Viola and M. J. Jones, Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), 1, I-511-I-518, 2001. https://doi.org/10.1109/CVPR.2001.990517

[28] I. Bamba, J. Yashika, Singh and P. Chawla, Face recognition techniques and implementation. Manav Rachna International Institute of Research and Studies, Faridabad, Haryana, India, 2022.

[29] M. Ganvir, A. Panchabhai, N. Sakhare, R. Thelkar and K. Wani, Face recognition using OpenCV. Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, Maharashtra, India, 2023.

[30] P. J. Phillips, et al, The FERET evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 10, pp. 1090-1104, 2000.

[31] W.Zhao, R. Chellappa, A. Rosenfeld and P. J. Phillips, Face recognition: A literature survey. ACM Computing Surveys, vol. 35, no. 4, pp. 399-458, 2003. https://doi.org/10.1145/954339.954342

[32] N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series. MIT Press, 1949

[33] C. Tomasi and R. Manduchi, Bilateral Filtering for Gray and Color Images. Proceedings of IEEE Conference on Computer Vision. IEEE, 1998. https://doi.org/10.1109/ICCV.1998.710815

[34] I. Sobel, An Isotropic 3x3 Image Gradient Operator. HPLABS, Measurement and Manufacturing Research Center. [Online]. Available: https://researchgate.net/publication/239398674_An_Isotropic_3x3_Image_Gradient_Operator.

[35] H. Bay, T. Tuytelaars, and L. Van Gol, SURF: Speeded Up Robust Features. European Conference on Computer Vision, 2006.

[36] D.G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, vol. 60, no. 2, 2004.

[37] Y. Taigman, M. Yang, and L. Wolf, DeepFace: Closing the Gap to Human-Level Performance in Face Verification. IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[38] N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[39] D. Aydemir, Revitalizing Turkish Mythological Elements through Artificial Intelligence Applications in Graphic Design: A Case Study on Midjourney, ISVOS, vol. 7, no. 2, pp. 187–205, 2023, https://doi.org/10.47897/bilmes.1400144.

[40] U. Saray and U. Çavdar, “Comparison of Different Optimization Algorithms in the Fashion MNIST Dataset”, IJMSIT, vol. 8, no. 2, pp. 52–58, 2024.

[41] V. Karaca and E. Yaşar, “Performing Distance Measurements Of Fixed Objects Detected With Yolo Using Web Camera”, ISVOS, c. 8, sy. 1, ss. 87–93, 2024, https://doi.org/10.47897/bilmes.1502873.

[42] F. Salahshoor (Director), Prophet Joseph [TV series]. IRIB, 2008.

[43] M.Sonka, V. Hlavac and R. Boyle, Image processing, analysis, and machine vision. Cengage Learning, 2014.

[44] Itseez. Open Source Computer Vision Library, 2015. [Online]. Available: https://opencv.org/.

[45] K. Fukunaga and L. D. Hostetler, The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. IEEE Transactions on Information Theory, vol. 21, no. 1, pp. 32-40, 1975. [Abstract]. Available: ProQuest, https://www.scirp.org/reference/referencespapers?referenceid=2651561.

[46] R. Lini, Different filters for image processing, 2021. [Online]. Available: https://medium.com/@rajilini/different-filters-for-image-processing-698e72924101.

[47] G. Bradski, The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.

[48] D. Comaniciu and P. Meer, Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, 2002. https://doi.org/10.1109/34.1000236.

[49] A. Majumdar and R. K. Ward, Robust Classifiers for Data Reduced via Random Projections. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 40, no. 5, pp. 1359-1371, 2010. https://doi.org/10.1109/TSMCB.2009.2038493.

[50] P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea, Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 11, pp. 1473-1488, 2008. [Abstract]. Available: ProQuest, https://ieeexplore.ieee.org/document/4633644.

[51] G. Özmen and R. Kandemir, Haar Dalgacıkları ve Kübik Bezier Eğrileri İle Yüz İfadesi Tespiti. ELECO '2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu, 2012.

[52] K. Kadir, M. K. Kamaruddin, H. Nasir, S. Safie, Z. Bakti, A comparative study between LBP and Haar-like features for Face Detection using OpenCV. ICE2T 4th International Conference on Engineering Technology and Technopreneuship, 2014.