BİLGİSAYAR BİLİMLERİ VE TEKNOLOJİLERİ DERGİSİ
JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGIES
http://dergipark.org.tr/tr/pub/bibted

Araştırma Makalesi

Örtülü Yüzlerin Tanınmasında Haar Cascade ve MongoDB Entegrasyonuyla Geliştirilen Yüz Tanıma Sisteminin (YTS) Performans Değerlendirmesi

Anıl YILDIZ[1], Zafer GÜNEY[2], Hakan AYDIN[*2]

[1]*İstanbul Gedik University, Department of Information Security Technology, Istanbul, 34020, Türkiye*
[2]*İstanbul Topkapı University, Department of Computer Engineering, Istanbul, 34020, Türkiye*

Anahtar Kelimeler:
Yüz tanıma teknolojisi
Maske takma
COVID-19 pandemisi
Haar Cascade yöntemi
MongoDB veritabanı

ÖZ

Her ne kadar geleneksel yüz tanıma sistemleri (YTS) maske takılıp takılmadığını belirli bir başarı oranında belirleyebilseler de maske takan kişilerin yüzlerinin büyük bir kısmının örtülü olması nedeniyle başarısız olabilmektedirler. Özellikle maske takan bireylerin yüzlerinin önemli bir kısmının örtülü olmasından kaynaklanan zorluklar mevcut YTS'lerin performansını sınırlamaktadır. Bu araştırmada maske takılmış yüz tanıma için Haar Cascade yönteminin OpenCV kütüphanesi kullanılarak gerçek zamanlı olarak MongoDB veritabanı ile entegrasyonun yapılması ve bu durumun kapsamlı deneylerle performansının ortaya konması amaçlanmıştır. Deneylerde maskeli yüzlerin büyük bir kısmının örtülü olduğu gerçekçi yüz görüntülerinden bu çalışma kapsamında oluşturulan veri seti kullanılmıştır. Araştırmamız, yüz tanıma doğruluğunun maskelenmiş yüzler için %85, maskesiz yüzler için %61 ve yüzün yarısı farklı bir nesne tarafından kapatıldığında %41 olduğunu göstermiştir. Bu çalışmanın Haar Cascade yöntemini gerçek zamanlı veritabanı yönetimi entegrasyonuyla birleştirerek daha etkili ve uygulanabilir bir maske tespit çözümü sunması bağlamında literatüre katkı sağlayacağı değerlendirilmektedir.

Performance Evaluation of Face Recognition System (FRS) Developed with Haar Cascade and MongoDB Integration in Recognition of Covered Faces

Keywords:
Face recognition
Mask wearing
COVID-19 pandemic
Haar Cascade method
MongoDB database

ABSTRACT

Although traditional face recognition systems (FRS) can detect with a certain success rate whether a mask is worn, they may fail because most of the faces of the people who wear masks are covered. The difficulties arising from the fact that a significant part of the faces of individuals wearing masks are covered limits the performance of existing FRSs. This research, it is aimed to integrate the Haar Cascade method with the MongoDB database in real-time using the OpenCV library for mask-wearing face recognition and to demonstrate its performance with extensive experiments. In the experiments, the data set created within the scope of this study from realistic face images, in which most of the masked faces are covered, was used. Our research has shown that the accuracy of face recognition is 85% for masked faces, 61% for unmasked faces, and 41% when half of the face is covered by a different object. It is considered that this study will contribute to the literature in terms of providing a more effective and applicable mask detection solution by combining the Haar Cascade method with real-time database management integration.

[*]**Sorumlu Yazar**

(anil.yildiz@gedik.edu.tr) ORCID 0000-0003-4607-6660
(zaferguney@topkapi.edu.tr) ORCID 0000-0003-1974-4264
[*](hakanaydin@topkapi.edu.tr) ORCID 0000-0002-0122-8512

e-ISSN: 2717-8579

Geliş Tarihi: 08/08/2023; Kabul Tarihi: 22/09/2023

1. INTRODUCTION

It can be observed that the COVID-19 pandemic, which is a significant health crisis affecting people worldwide, is still ongoing in different regions and different ways today. This outbreak has been experienced and learned as a health crisis that affects the health of all people worldwide. COVID-19 is a respiratory illness caused by a coronavirus called SARS-CoV-2. It is a disease in which the virus spreads through respiratory droplets when an infected person coughs, sneezes, or talks. Another important aspect that has been experienced is that the course of this outbreak can vary depending on the measures taken by countries. One of these measures is the wearing of masks by people, which has been experienced and learned as another important aspect. Wearing masks is essentially an effective measure against COVID-19 and has emerged as another significant issue during this pandemic. The World Health Organization (WHO) and other health authorities emphasize that wearing masks plays an important role in reducing the risk of infection. Wearing a mask not only reduces the risk of transmitting the disease by preventing the spread of respiratory droplets in the air but also reduces the risk of inhaling respiratory droplets from others and getting infected. The importance of wearing masks is particularly emphasized in indoor or crowded environments. Masks should be used to protect both yourself and others around you. Furthermore, wearing masks, along with other measures such as hand hygiene and social distancing, can further reduce the risk of infection. Measures to be taken against COVID-19 include social distancing, mask usage, hygiene practices, limitation of mass gatherings, testing and monitoring measures, and travel restrictions (Ciotti et al., 2020). It has been observed that wearing masks is an important strategy in controlling the spread of COVID-19 during the pandemic and that mask usage reduces the risk of infection and transmission by reducing the spread of respiratory droplets. Additionally, during this global pandemic, it has been learned through experience that mask types, proper mask usage, and wearing masks are crucial in the fight against this disease. Particularly, it has been experienced and learned that mask usage is important in public areas and situations involving contact with infected individuals, and it contributes significantly to controlling the outbreak. The detection of whether masks are worn or not is of great importance in controlling the outbreak and ensuring safety in public areas.

Face recognition systems (FRS) are technologies that perform authentication or recognition by analyzing the characteristics of people's faces. Various techniques are currently being developed, including local, holistic, and hybrid approaches, which define a face image using only a few or all of the facial image features (Kortil et al., 2020). Face recognition involves detecting faces in an image or video and associating these faces with specific people (Parmar et al., 2014). These systems typically use an image taken from a person's face via a camera or image sensor. Face recognition algorithms mathematically analyze the unique features of the face, such as the shape of the face, eye position, nose structure, and lips, and create a unique facial signature for a person. This signature is then compared with other recorded facial images for matching and authentication or recognition is performed. Face recognition involves a series of processes and analyses carried out to detect faces in an image or video, extract unique features of faces, and identify or verify a particular person using these features (Grudin, 2000). Face recognition usually consists of two stages: face detection, which is the process of detecting faces in an image or video, and face recognition, which involves matching and identifying specific people (Parmar et al., 2014). Facial recognition systems are used in various applications such as security, authentication, access control, and crime prevention. Research is currently being conducted on facial recognition technology in connection with the topic of wearing masks. Traditional face recognition systems perform authentication or recognition of individuals by analyzing the characteristic features of their faces. However, since most of the face is covered, the accuracy and effectiveness of these systems may decrease when dealing with people wearing masks. For this reason, it is necessary to develop new approaches to update face recognition systems or detect mask-wearing. In some studies, different algorithms and methods have been used to detect whether a mask is worn. Various studies are being carried out to enable facial recognition systems to produce more accurate results for individuals who currently wear masks and to detect mask-wearing. During the COVID-19 pandemic, facial recognition technology has become an important tool for controlling the spread of infection and maintaining security in public spaces (Van Natta et al., 2020). Facial recognition technology is important during the COVID-19 pandemic because it can detect mask-wearing, helping to control the spread of infection and ensure safety in public areas (Mundial et al., 2020). In this context, the detection of mask use by using face recognition systems has recently become a research and examination subject.

This study aims to integrate the Haar Cascade method with the OpenCV library and integrate it with the MongoDB database in real-time for accurate recognition of masked faces, and to evaluate the performance of this approach with extensive experiments and sample application scenarios. Python, MongoDB, and OpenCV, which are the subject of this study, are popular technologies used in different but compatible areas. Python is a user-friendly and easy-to-understand programming language that is widely used to develop various applications thanks to its extensive library support. OpenCV is a library for image and video processing applications. It integrates with Python and supports a wide variety of visual analytics such as image processing, object detection, face recognition, and video analytics. Thus, it provides powerful tools for manipulating, analyzing, and manipulating image and video data. Experiments carried out within the scope of the study were carried out using a dataset created from realistic facial images, taking into account different lighting conditions. This dataset aims to more robustly evaluate the effectiveness of the proposed method by including cases where most of the masked faces are covered. MongoDB is a document-based NoSQL database that provides a scalable, flexible, and high-performance data storage solution. It works with unstructured data models and is ideal for processing large datasets. This database is also equipped with features such as high-performance query processing, backup, and replication. The obtained results confirm that the proposed method can be used successfully in real-time applications and its effectiveness in accurately recognizing masked faces. Although our research has decreased, it offers an innovative solution for the detection of mask-wearing, which continues to be an important need within the scope of the COVID-19 process, which still has an impact in many parts of the world. The method proposed in our study can be used in areas such as security, health, and public space management and contributes to the development of relevant research and applications in these areas. In addition, it is considered that the applied approach presented in our study on the compatible use of Python and MongoDB will be a reference for researchers and developers using these technologies.

2. METHOD

This study addresses the existing limitations in recognizing obscured faces, particularly focusing on the challenges posed by a significant portion of the face being concealed due to mask-wearing. The aim is to overcome these challenges and develop a system capable of effectively recognizing masked faces. To achieve this goal, the Haar Cascade method was implemented using the OpenCV library, and simultaneous integration of recognized facial data into a MongoDB database was accomplished. During the data collection phase, a dataset representative of real-world scenarios involving diverse degrees of face masking was curated. The Haar Cascade method was selected as the preferred algorithm for facial detection and extensively evaluated using the compiled dataset. During this phase, performance metrics such as facial detection accuracy, false positives/negatives, and processing times were meticulously analyzed. Additionally, through the integration with the MongoDB database, the real-time storage and management of recognized facial data were realized. Performance evaluation was conducted rigorously on the curated dataset, measuring the recognition efficacy of the Haar Cascade method against various performance criteria. These criteria encompassed accuracy rates, erroneous positive and negative outcomes, and processing speeds.

The work's summary is depicted in **Figure 1**, which presents the main steps. First, camera images are captured in real time. Then, facial recognition is carried out using Haar Cascade and OpenCV. The goal is to accurately identify faces, especially those wearing masks. Next, recognized face data is stored in a MongoDB database for future access. Lastly, the system's performance is evaluated through experiments to analyze accuracy, speed, and reliability impacts.

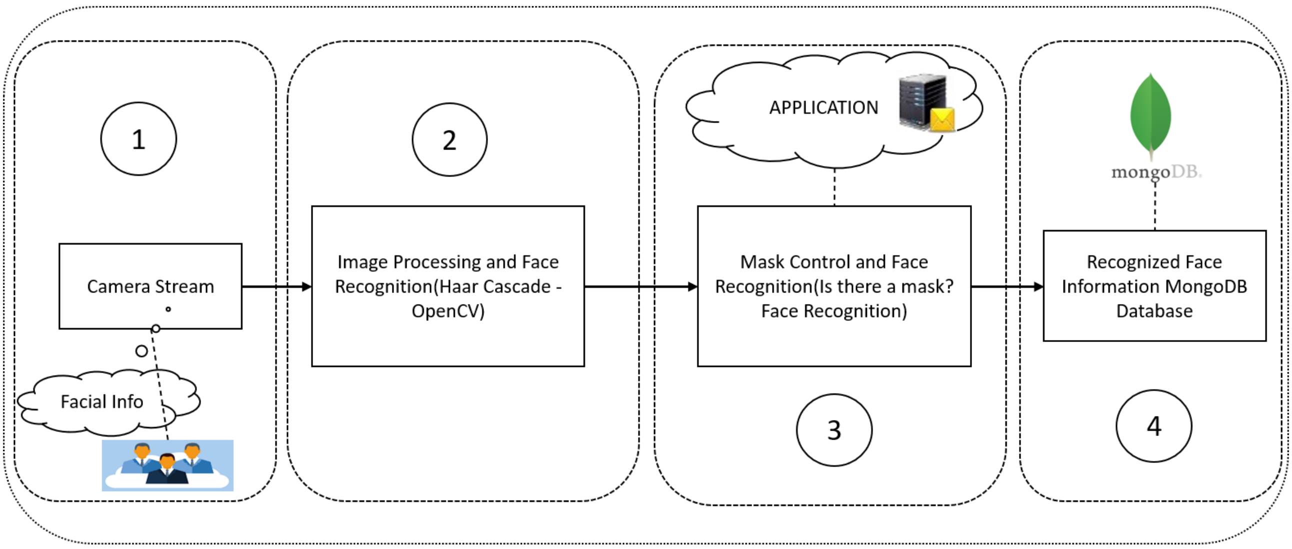

Figure 1. The block diagram summarizing the study

The system's performance was validated through illustrative real-world applications, effectively simulating real-time conditions. The results obtained from experiments were presented with thorough analysis and comparative discussions. The contributions of this study to the scientific discourse and its potential applications were emphasized. In particular, the amalgamation of the Haar Cascade method with MongoDB integration was deemed to hold the potential to yield an enhanced solution for effective mask detection, offering an avenue for practical applications.

3. RELATED WORKS

In the study conducted by Mundial et al. (2020), the facial recognition problem during the COVID-19 pandemic was addressed. The authors presented information about the challenges faced by facial recognition technologies in accurately identifying individuals wearing masks. The article discusses different approaches, methods, and techniques for recognizing individuals wearing masks and focuses on the impact of mask-wearing on facial recognition systems. Additionally, the article addresses the security, privacy, and ethical issues that have emerged with the increased use of facial recognition technologies during the pandemic. It provides insights into the current challenges and future potential of facial recognition technologies in the context of the COVID-19 period. In the study conducted by Karadağ (2020), face recognition was performed by detecting the face with the Haar-Cascades classifier and then using the LBPH (Local Binary Patterns Histograms) method. Van Natta et al. (2020) examined the rise and regulation of thermal facial recognition technology during the COVID-19 pandemic. The authors studied the increasing use of thermal facial recognition technology, its applications in security and healthcare sectors, ethical and privacy concerns, and regulatory measures. They also discussed the potential benefits and risks of facial recognition technology. Vansh et al. (2020) discussed a face recognition method developed using the "YCbCr" and "Adaboost" algorithms. The article states that the use of the YCbCr color space enhances face detection performance, and the AdaBoost algorithm improves classification accuracy. This study presents a new approach to enhance the performance of face recognition systems and achieve more accurate and efficient results. In the study conducted by Özdemir and Koç (2019), emotion recognition was performed using deep learning methods for seven different facial expressions with a dataset (RidNet) created using publicly accessible images from the internet. Then, transfer learning was performed over RidNet with well-known convolutional neural network architectures in the literature such as AlexNet, GoogLeNet, and ResNet101. The Compound Facial Expressions of Emotion (CE) and Static Facial Expressions in the Wild (SFEW) datasets were designated as test datasets. Firstly, the convolutional neural network architecture, which shows the best classification performance, was determined by experimental studies. This convolutional neural network is trained with AffectNet, The Karolinska Directed Emotional Faces (KDEF), and RidNet. Similar classification performances were obtained when networks trained with AffectNet, KDEF, and RidNet were tested with a dataset (CE) created in a controlled environment. In the test dataset (SFEW) in the uncontrolled environment, the network trained with RidNet provided a significant advantage over other networks. Kekül et al. (2018) compared Eigenfaces and Artificial Neural Networks in face recognition applications. The article highlights the effectiveness and different advantages of both methods in face recognition performance. The Eigenfaces method stands out for its fast operation and low computational requirements, while artificial neural networks are capable of detecting more complex features and providing higher recognition accuracy. The article provides a detailed comparison of these methods in terms of their applications, advantages, and disadvantages. It emphasizes that this study provides a guiding framework for the decision-making process in face recognition systems and suggests that better results can be achieved by employing different methods. Mamak et al. (2020) researched a real-time face recognition-based personnel control and monitoring system. The article explains how facial recognition technology can be used in personnel control and monitoring processes and describes the functioning of the designed system. The designed system is used for real-time facial recognition in applications such as personnel records, attendance tracking, and access control. The article also provides information about the techniques and algorithms used in the system design. Atasoy and Tabak (2018) developed a face recognition application using Support Vector Machines (SVM). The article extensively discusses how the SVM algorithm can be applied in the field of face recognition, including its working principles and application stages. The process of data collection for face recognition, feature extraction, training, and classification steps are emphasized. The article provides information about the performance, accuracy rate, and advantages of SVM in face recognition, along with sharing the results of experiments conducted on different datasets. In the study conducted by Terzopoulos and Waters (1990), the analysis of facial images using physical and anatomical models was carried out. The article focuses on how facial recognition systems can be based on physical and anatomical models to obtain more accurate and reliable results. It is stated that physical models represent the structure and movement of the face with mathematical equations and are used in the analysis of facial movements. Anatomical models, on the other hand, include databases representing the anatomical structures of the face and feature-based analysis methods. The article presents research and results on how these models can be used to improve facial recognition performance and their effects on the analysis process. In the study conducted by Kirby and Sirovich (1990), a method was developed using the Karhunen-Loeve procedure for characterizing human faces. Rodriguez and Marcel (2006) performed a study on face verification using aligned local binary pattern histograms. In the study conducted by Ngo et al. (2009), a modular approach was used to implement an FPGA-based design for a real-time face detection system. These studies, examined within the scope of our research, indicate that during the COVID-19 pandemic, research was conducted on the challenges of accurately recognizing individuals wearing masks using facial recognition technology and addressing the security, privacy, and ethical issues of this technology. It is also understood that studies were conducted on enhancing the performance of facial recognition systems through the use of different methods and developing FPGA-based designs for real-time applications.

In our research, studies on the Python programming language and MongoDB database were also examined. In the study conducted by Araujo et al. (2021), an analysis comparing the performance of NoSQL Cassandra and MongoDB databases was conducted. The book "Programming Python" by Lutz (2010) focuses on the Python programming language and object-oriented programming. The study conducted by Lemenkova (2020) discusses the use of Python libraries such as matplotlib, seaborn, and pandas for visualizing geographic datasets generated by QGIS. In Taylan's (2014) article, it is stated that Python is the most commonly taught programming language in universities in the United States. Kolakowski (2020) provides information about the popularity of Python among the popular programming languages on GitHub and its ranking on the platform. Saran and Saran (2019) conducted a comparative security analysis of NoSQL databases such as Cassandra and MongoDB. In the study by Daşdemir and Kara (2019), the performance of MongoDB, a NoSQL database, was analyzed under different workloads. The study presents information on evaluating MongoDB's performance based on various metrics and criteria, and how it behaves under different workloads. The book "MongoDB" defines MongoDB as a document-oriented NoSQL database. The book extensively covers MongoDB's features, data modeling, query language (MongoDB Query Language - MQL), and other related topics (Mongo, 2015). The book "MongoDB in Action: Covers MongoDB Version 3.0" also defines MongoDB as a document-oriented NoSQL database. The book highlights MongoDB's flexible data model, high-performance query capabilities, horizontal scalability features, and other advantages (Banker et al., 2016). In the study "MongoDB vs Oracle--database comparison," MongoDB is also defined as a document-oriented NoSQL database. The study evaluates MongoDB's flexible data model, high performance, scalability, and ease of data management by comparing it with the Oracle database (Boicea et al., 2012). The book "MongoDB Data Modeling" published by França (2015) focuses on the topic of data modeling in MongoDB. The book explains the concept of NoSQL databases and specifically discusses data modeling strategies for MongoDB. It explores how the data structure in document-oriented databases should be designed, different approaches from relational databases, and best practices. Additionally, the book provides detailed explanations of MongoDB's data modeling tools and techniques. It emphasizes that MongoDB is a document-oriented NoSQL database and can be used for data modeling strategies. In the studies examined within the scope of our research, the studies highlight the need to consider security, privacy, and ethical issues with the increased use of facial recognition technology during the pandemic. The examined studies reveal that research has been conducted on developing new algorithms and methods to improve the performance of facial recognition systems, the widespread use of thermal facial recognition technology, and the utilization of different databases and feature-based analysis methods.

4. THE PROPOSED SYSTEM

This section outlines the new system we have developed to improve the recognition of faces when partially covered by masks. To achieve this, we combined the Haar Cascade method with the MongoDB integration. The system we propose has two main parts: the part that recognizes faces using the Haar Cascade method, and the part that helps store and manage data using MongoDB. The facial recognition part uses the Haar Cascade method, which is a well-known way of identifying different parts of the face such as the eyes and nose. This helps you find faces in real-time, even if they are partially covered. This part can handle different lighting conditions and how much of the face is covered, making it good for real situations. The MongoDB part allows us to store and manage the face data we find. This includes recording pictures of faces we recognize and details such as when we saw them and how confident we were that they were correct. This way, we can keep everything up to date in real-time and make sure that the data is consistent and ready for future use. The system works step by step to achieve its goals. Initially, it takes live images or video feeds where faces can be hidden by masks. Using the Haar Cascade method, the system identifies faces in these images by precisely positioning facial features, even if they are partially hidden. Recognized faces are then compared to known profiles. Once a match is identified, the system seamlessly integrates the recognized faces and related data into the MongoDB database in real-time. This database integration is crucial in maintaining data accuracy and consistency. The MongoDB component plays a crucial role in effectively organizing and managing face data, enabling fast queries, data analysis, and future enhancements to the face recognition model. This comprehensive process leverages the power of Haar Cascade and MongoDB to improve facial recognition performance while adapting to real-world scenarios, ultimately contributing to a practical and efficient system.

5. EXPERIMENTAL STUDIES

The experiments included various levels of covering of masked faces under different light conditions based on real-world scenarios. The dataset used for the experiments includes masked face images that reflect real-world conditions. These images represent a wide range of occlusion levels and various light conditions. The data set is enriched with the necessary labels to objectively evaluate the detection results on each face image. Experiments were meticulously conducted under different predetermined occlusion levels and various light conditions. Each experiment was performed on a dataset containing a wide range of facial images. Face detection was performed on each image using the Haar Cascade method. Experimental results were rigorously evaluated using comprehensive performance metrics such as detection success for each image, false positive/negative results, and processing times. The detailed results of these experiments helped us to understand how effective the developed face recognition system is in different covering levels and light conditions. The data obtained constitute an important source of information in terms of identifying weak points in the system and identifying areas where future developments can focus.

In our study, firstly, Python was downloaded from the official website and installed on the computer. Then, the OpenCV library was installed for facial recognition and image processing. OpenCV was downloaded and installed from the Python package manager using pip. MongoDB database was used for data storage and management. MongoDB Community Edition was downloaded from the official website and installed. MongoDB, a NoSQL-based database, was preferred for storing and managing the results obtained in the study. The Haar Cascade method is a feature-based classification algorithm used for image processing and object detection. This method utilizes a pre-trained classifier to recognize specific objects. Haar Cascade consists of a step-by-step process involving feature extraction and classification stages. The Haar Cascade method is based on Haar-like features that define different characteristics of an object, such as its horizontal edges, vertical edges, corners, etc., represented by black and white rectangular regions. These features are applied to the image through sliding and scaling operations. Using a pre-trained classifier, training is performed on a dataset containing positive and negative examples. Positive examples represent images of the desired object, while negative examples represent images without the desired object. The training process involves adjusting the classifier to differentiate the desired object based on these features. During the object detection stage, the trained classifier scans the image to detect potential object locations. These detected regions are then scaled to a higher level of detail for further examination. As a result, the identified objects are recognized with a certain level of precision and accuracy. The Haar Cascade method is widely used in various applications, including face recognition, object detection, and even eye tracking. Pre-trained classifiers can often be found in image processing libraries such as OpenCV and can be easily implemented using these libraries. Haar Cascade is known as a fast and effective object detection method for real-time applications.

Table 1 shows the setting parameters used at different stages of the study and their values. In the study, the adjustment parameters and values of the Haar Cascade method used for face detection are given.

Table 1. Setting Parameters and Values

Method and Purpose	Parameters and Values
Face Detection with Haar Cascade	Scale Factor: 1.1 Min Neighbors: 3 Min size: (30, 30)
MongoDB Integration and Database	Link URL: mongodb://localhost:27017 Username: user123 Password: password123
Performance Evaluation	Recognition Accuracy: 85% Speed: 0.25 seconds/frame False Positive Rate: 10% False Negative Rate: 5%

According to the information presented in the table, the Scale Factor parameter determines how much the picture jumps at each scale level and the value used here is 1.1. The Min Neighbors parameter indicates how many neighbors are required for each face region during detection, and its value is 3. The Min Size parameter refers to the minimum size of the face to be detected and is indicated here as (30, 30). Under the heading "mongoDB Integration and Database", the setting parameters and values used in the real-time database integration of the study are presented. The link URL parameter refers to the URL used to connect to the MongoDB server and is given as an example, "mongodb://localhost:27017". The username parameter represents the username used to access the database and is designated "user123". The password parameter contains the password of the specified user name and is shown as "password123". Under the title of "Performance Evaluation", the metrics used to evaluate the performance of the developed system and the results obtained are explained. The Recognition Accuracy metric reflects the accuracy of recognized faces of the developed system and is shown here as 85%. The speed metric expresses the average time for the recognition process of each image and its value is stated as 0.25 seconds/frame. The False Positive Rate metric reflects the percentage of system false positive results and is given here as 10%. The False Negative Rate metric represents the percentage of faces that should be recognized but not mistakenly recognized, and its value is 5%.

As part of the experimental steps, a dataset consisting of facial photographs of individuals wearing and not wearing masks was prepared for facial recognition study. An example image from the dataset is shown in **Figure 2**.

Figure 2. An example image used from the dataset

Face detection and preprocessing steps were performed using OpenCV. The face detection algorithm of OpenCV was used to detect faces. In this step, faces were cropped and resized, and necessary preprocessing steps were applied. The Haar Cascade method was used in the study. It can detect faces quickly and effectively by using predefined features with a pre-trained classifier. However, it may occasionally produce false positive or false negative results. It may provide less accuracy, especially in cases where faces vary in angles, positions, and sizes. Therefore, this method was preferred in our study. In the face recognition step, pre-trained face recognition models in OpenCV were used. These models include algorithms used to recognize faces. Pre-trained models in OpenCV for face recognition were used by employing the Haar Cascade method. In this context, first, the OpenCV library was included in our project. Then, the Haar Cascade model was loaded for face recognition. OpenCV provides pre-trained Haar Cascade files. The "haarcascade_frontalface_default.xml" file is a commonly used model for face detection. Afterward, images were loaded into the system for face detection. The system status regarding an image of a person without a mask is shown in **Figure 3**.

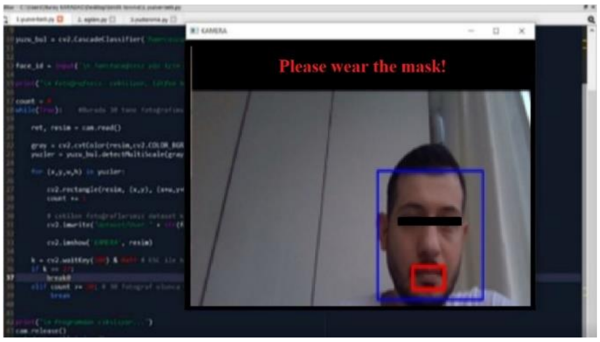

Figure 3. An Unmasked Image

In **Figure 4**, the system status regarding an image of a person wearing a mask is shown

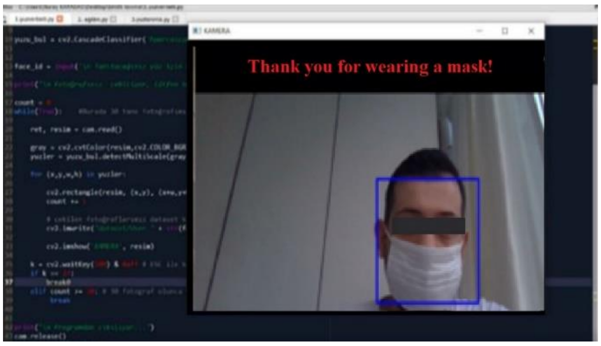

Figure 4. A Masked Image

These pieces of information were stored in the MongoDB database to preserve and manage the results. A suitable MongoDB driver in Python was used to save the results in MongoDB. The outcomes of the experimental studies were documented as records and then stored in the MongoDB database. In the software code utilized for face recognition, the 'detectMultiScale' function was employed to detect faces in grayscale images. The 'scaleFactor' controls the reduction rate of the image scale, while 'minNeighbors' indicates the required number of neighbors for each candidate rectangle. 'minSize' specifies the minimum size of a face. Finally, the results obtained are stored in the database. Each image contains the identity of the detected face, whether a mask is worn or not, and other relevant data. In this way, the necessary steps and experimental processes for the face recognition study conducted within the scope of the research have been completed. The face recognition process was carried out using Python, OpenCV, and MongoDB, and the results were stored in the MongoDB database. The system state of the image of a person without a mask with half of the face covered is shown in **Figure 5**.

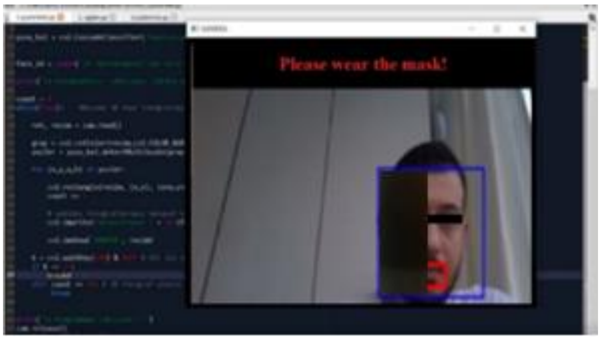

Figure 5. A Half Face Covered Image

6. RESULTS AND DISCUSSION

The operations conducted within the scope of the experimental studies are presented in **Table 2.**

Table 2. Experimental Results

Experiments	Explanation
Experiment 1: Preparation of the dataset.	A dataset consisting of face photographs of individuals wearing and not wearing masks was prepared. Each photograph had a label indicating whether a mask was worn or not.
Experiment 2: Face detection and preprocessing steps.	Face detection and preprocessing steps were performed using OpenCV. The face detection algorithm in OpenCV was used to detect faces.
Experiment 3: Utilization of the Haar Cascade method for mask classification	This method was used to determine whether faces had masks or not.
Experiment-4: Utilization of pre-trained models for face recognition	The pre-trained face recognition models in OpenCV were used, which include algorithms for recognizing faces.
Experiment 5: Creation of the database and assignment of identities	A database was created for the face recognition process, and each person was assigned an identity (ID).
Experiment-6: Determination of identities for detected faces	The detected faces were compared with the faces in the database to determine their identities.
Experiment-7: Saving the results to the MongoDB database	The results were saved to the MongoDB database using a suitable Python driver.

In the first experimental phase, a data set was created for face recognition and mask classification processes. This dataset includes face photographs of individuals wearing and not wearing masks. There are tags on each photograph indicating whether the person concerned is wearing a mask or not. In the second experimental phase, face detection and preprocessing steps were performed. At this stage, face detection was performed using the OpenCV library. OpenCV's powerful face detection algorithm has been used to successfully detect faces in the dataset. In the third experiment, the Haar Cascade method was used to determine whether there were masks on the faces. This method can detect the presence of a mask by detecting different features on the face. In the fourth experiment, face recognition was performed using pre-trained face recognition models in OpenCV. These models include various algorithms for recognizing faces, and the purpose of the experiment is to evaluate the performance of these models. In the fifth step, a database was created for face recognition, and a unique identity (ID) was assigned to each individual. This step is important for associating and tracking recognizable faces. In the sixth experiment, the detected faces were compared with the faces in the database, and their identities were determined. This stage ensures that the accuracy and reliability of the face recognition algorithm are tested. In the seventh experiment, the results obtained were saved in the MongoDB database using a suitable Python driver. This step ensures that the results are stored and made accessible for later analysis.

In the experiments conducted shown in **Figure 6**, the average success rate of "Perception of a Single Person's Face While Masked", "Perception of a Single Person's Face While Unmasked" and "Perception of Half of the Face While Covered with a Different Object" was also obtained.

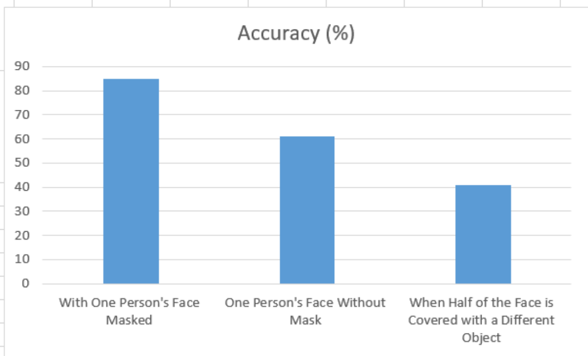

Figure 6. Evaluation of performed experiments

The experiment involving the recognition of a single individual's face while wearing a mask resulted in an accuracy of 85%. This finding signifies the system's capability to effectively identify faces even when a mask is worn, achieving a commendable accuracy level. When assessing unmasked faces, the accuracy remained at 61%, indicating that the recognition of faces without masks also demonstrated a substantial level of success. On a different note, the experiment where half of the face was concealed with a distinct object yielded an accuracy rate of 41%. This outcome implies that recognizing faces in cases where a substantial portion is hidden encounters challenges, significantly impacting the system's accuracy.

The experimental results we obtained in the study were also compared with the results of similar studies in the literature (**Table 3**).

Table 3 Evaluation of performed experiments

State	Accuracy (%)
With One Person's Face Masked (This Study)	85
One Person Face Masked (Karadağ, 2020)	84.5
While One Person's Face Is Masked (Özdemir, 2019)	80
While One Person's Face Is Masked (Atasoy, 2018)	80

As shown in Table 3, the success rate of 85% obtained as a result of this study is slightly lower than the success rate of 84.5% in the study of Karadağ (2020). While Karadağ's study showed that he was able to achieve slightly higher accuracy in recognizing masked faces, this study still achieved an acceptable level of recognition success of 85%. On the other hand, Özdemir's (2019) study also achieved 80% success in recognizing masked faces. This is a result quite similar to the success rate in this study. Similarly, 80% success was achieved in Atasoy's (2018) study. While these two studies had similar success levels in recognizing masked faces, this study focused on more complex situations and achieved an 85% success rate. In conclusion, this study performs competitively in the recognition of masked faces when compared to other similar studies. While the work of Karadağ has slightly higher success, a similar level of success is shown with the work of Özdemir and Atasoy. This comparison helps us to evaluate the masked face recognition capabilities of the study from a broader perspective.

This series of experiments was carried out to evaluate the performance of different methods in face recognition and mask classification problems. The experimental results reveal the advantages and difficulties of each method. The effectiveness of pre-trained models in face recognition and the success of the Haar Cascade method in mask detection are interesting. This study provides important information that can be used in a wide area from security applications to biometric recognition systems.

7. CONCLUSION

The results of this study reveal the limitations of traditional face recognition systems when individuals wearing masks have their faces covered. In particular, the fact that the faces of people wearing masks are largely covered significantly reduces the accuracy of existing systems. The research presents an approach that enables real-time recognition of masked faces using the Haar Cascade method and combines this with MongoDB database integration. Experiments show that the proposed method is effective even under different lighting conditions and is suitable for real-world applications. It is considered that this study will contribute to the literature within the scope of its potential to offer a more effective solution, especially by addressing the difficulties in mask detection.

KAYNAKÇA

Araujo, J. M. A., de Moura, A. C. E., da Silva, S. L. B., Holanda, M., de Oliveira Ribeiro, E., da Silva, G. L. (2021, June). Comparative performance analysis of NoSQL Cassandra and MongoDB databases. In 2021 16th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). IEEE.

Atasoy, N. A., Tabak, D. (2018). Destek vektör makineleri kullanarak yüz tanıma uygulaması geliştirilmesi. Engineering Sciences, 13(2), 119-127.

Banker, K., Garrett, D., Bakkum, P., Verch, S. (2016). MongoDB in action: covers MongoDB version 3.0. Simon and Schuster.

Boicea, A., Radulescu, F., Agapin, L. I. (2012, September). MongoDB vs Oracle database comparison. In 2012 third international conference on emerging intelligent data and web technologies (pp. 330-335). IEEE.

Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W. C., Wang, C. B., Bernardini, S. (2020). The COVID-19 pandemic. Critical reviews in clinical laboratory sciences, 57(6), 365-388.

Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM) (pp. 210-214). IEEE.

da Rocha França, W. (2015). MongoDB data modeling. Packt Publishing Limited.

Daşdemir, Y., Kara, B. C. (2019). Farklı iş yükleri altında NoSQL sistemlerinin performans analizi. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 8(4), 1466-1477.

Mongo, D. B. (2015). Mongodb.

Grudin, M. A. (2000). On internal representations in face recognition systems. Pattern recognition, 33(7), 1161-1177.

Karadağ, N., Çetinkaya, A., & Aydın, H. (2020). Yerel İkili Desenler Histogramları ile Covid-19 Tanılı Kişiler Üzerinde Kimlik Analizi ve Bildiri Sistemi. Gazi Mühendislik Bilimleri Dergisi, 6(3), 172-183.

Kekül, H., Bircan, H., Arslan, H. (2018). Yüz tanıma uygulamalarında özyüzler ve yapay sinir ağlarının karşılaştırılması. Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi, 2(1), 51-59.

Kirby, M., Sirovich, L. (1990). Application of the Karhunen-Loeve procedure for the human faces. IEEE Trans. Pattern Analysis and Machine Intelligence, characterization, 12(1), 103-108.

Kolakowski, (2020, December, 3), 10 Most Popular Programming Languages on GitHub, GitHub Language Rankings, 2018-2020 (Accessed date:

October 19, 2021) https://insights.dice.com/2020/12/03/10-most-popular-programming-languages-on-github/

Kortli, Y., Jridi, M., Al Falou, A., Atri, M. (2020). Face recognition systems: A survey. Sensors, 20(2), 342.

Lemenkova, P. (2020). Python libraries matplotlib, seaborn and pandas for visualization geospatial datasets generated by QGIS. Analele stiintifice ale Universitatii" Alexandru Ioan Cuza" din Iasi-seria Geografie, 64(1), 13-32.

Lutz, M. (2010). Programming Python: powerful object-oriented programming. " O'Reilly Media, Inc.".

Mamak, U., Konyar, M. Z., Solak, S., Uçar, M. H. (2020). Gerçek zamanlı yüz tanıma tabanlı personel kontrol ve takip sistemi tasarımı. Avrupa Bilim ve Teknoloji Dergisi, (19), 497-504.

Mundial, I. Q., Hassan, M. S. U., Tiwana, M. I., Qureshi, W. S., Alanazi, E. (2020, September). Towards facial recognition problem in COVID-19 pandemic. In 2020 4th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM) (pp. 210-214). IEEE.

Ngo, H. T., Rakvic, R. N., Broussard, R. P., Ives, R. W. (2009, May). An FPGA-based design of a modular approach for integral images in a real-time face detection system. In Mobile Multimedia/Image Processing, Security, and Applications 2009 (Vol. 7351, pp. 83-92). SPIE.

Özdemir, R., & Mehmet, K. O. Ç. (2019). Yeni bir veri kümesi (RidNet) kullanarak kontrolsüz ortamda yüz ifadesi tanımanın derin öğrenme yöntemleri ile iyileştirilmesi. Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi, 6(2), 384-396.

Parmar, D. N., Mehta, B. B. (2014). Face recognition methods & applications. arXiv preprint arXiv:1403.0485.

Saran, M., Saran, N. (2019). Cassandra ve MongoDB NoSQL Veri Tabanlarının Karşılaştırmalı Güvenlik Analizi. Uluslararası Bilgi Güvenliği Mühendisliği Dergisi, 5(2), 1-11.

Terzopoulos, D., Waters, K. (1990, January). Analysis of facial images using physical and anatomical models. In Proceedings Third International Conference on Computer Vision (pp. 727-728). IEEE Computer Society.

Rodriguez, Y., Marcel, S. (2006). Face authentication using adapted local binary pattern histograms. In Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV (pp. 321-332). Springer Berlin Heidelberg.

Taylan, E. (2014), ABD'deki üniversitelerde en çok öğretilen programlama dili: Python Erişim Tarihi:19-10-2021.

Van Natta, M., Chen, P., Herbek, S., Jain, R., Kastelic, N., Katz, E., Vattikonda, N. (2020). The rise and regulation of thermal facial recognition technology during the COVID-19 pandemic. Journal of Law and the Biosciences, 7(1), lsaa038.

Vansh, V., Chandrasekhar, K., Anil, C. R., Sahu, S. S. (2020). Improved face detection using YCbCr and Adaboost. In Computational Intelligence in Data Mining: Proceedings of the International Conference on ICCIDM 2018 (pp. 689-699). Springer Singapore.