Republic of the Philippines
Western Mindanao State University
College of Computing Studies
DEPARTMENT OF COMPUTER SCIENCE
Zamboanga City

Enhanced Security and Log System For Western Mindanao State University (WMSU) Using Real-Time Face Recognition System

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Presented to the Faculty of
Department of Computer Science
College of Computing Studies

Ronald Dale A. Fuentebella
Josua Z. Habil
Researchers

Mr. Jaydee C. Ballaho
Adviser

April 2022

Western Mindanao State University

College of Computing Studies

DEPARTMENT OF COMPUTER SCIENCE

Zamboanga City

Approval Sheet

The Thesis attached hereto, entitled **"Enhanced Security and Log System for Western Mindanao State University (WMSU) Using Real-Time Face Recognition System"**, prepared and submitted by Ronald Dale A. Fuentebella and Josua Z. Habil, in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science, is hereby **recommended for Oral Examination**.

MR. JAYDEE C. BALLAHO
Adviser

APPROVED by the Oral Examination Committee on _____ with a rating of **PASSED**.

MS. LUCY F. SADIWA, MSCS
Chairperson

MR. SALIMAR B. TAHIL, MEnggEd
Member

ENGR. MARJORIE A. ROJAS
Member

ACCEPTED in partial fulfilment of the requirements for the degree of **Bachelor of Science in Computer Science**

ODON A. MARAVILLAS JR., MSCS
Head, Department of Computer Science

RODERICK P. GO, PhD
Dean, College of Computing Studies

Acknowledgment

We are pleased to have this opportunity to think all the people who are part of our journey in completing this thesis project. We are thankful to Engr. Marvic Lines our thesis 1 adviser who guides us in choosing the right study for our thesis paper.

We are most grateful to Mr. Jaydee Ballaho our thesis 2 adviser whom we consider our mentor for sharing his knowledge and expertise regarding object detection. We can say that without him this study will not be possible, he guides us in every step we take and point us in the right direction. He is the best teacher for he is loyal to his duties as an adviser, he would meet us every week just to check our progress and for that, we are very thankful.

To the panel members Mr. Salimar B. Tahil, Ms. Lucy Felix-Sadiwa, and Ms. Marjorie A. Rojas we thank you for your honest critique of our research, without it we would not be guided.

We would also like to thank our classmates and very good friends Mr. Jayson F. Beltran, Mr. Wenefredo P. Tejero, Mr. Emmanuel L. Toledo, and Mr. Mark Anthony D. Tubat for their honest opinions and criticism about the design elements of the system and for helping us with the proofreading and data gathering of our project.

This thesis project would not have been possible without the unending support of our colleagues, friends, classmates, and of course our family who serves as our inspiration in completing this paper.

We would also like to express our gratitude to all the people who participated and volunteered in the implementation of our system for without them we would have not gathered results.

And most importantly to our creator God almighty for the struggles and challenges that he put in our way that make us strong and persevere more for the success of this paper. God is always there to remind us that in achieving something there must be hardships for if we overcome this hardship the results and prize are sweet as honey

Abstract

Different biometrics are emerging because of their usefulness and fast identity recognition. WMSU is currently not using biometric technology to check personnel entry into the school campus, which is a hassle for visitors, students, and school employees. These resulted in developing a face recognition system capable of collecting unique facial features of an individual, converting the facial feature into a unique histogram, storing the facial feature, and recognizing the registered face in the system using the Haar Cascade Local Binary Pattern Histogram. Haar Cascade for detecting facial features and Local Binary Pattern Histogram for face recognition. Integrating these two algorithms results in a massive increase in facial recognition confidence rate up to 90% for both artificial and natural lighting conditions. Implementation of the system was done by imitating the main entrance of the school campus with the device for facial recognition set up, such as the camera and computer. Upon testing the system, it could register a subject for less than a minute and detect and recognize registered subjects almost instantly. The system can also detect unregistered subjects, assuring the security of all entry on the campus. The results suggest that facial recognition technology is effective as an alternative way for a biometric log system for WMSU. In conclusion, it satisfactorily achieves its objectives and performs well in its implementation.

Keywords:

Face Recognition, Face Detection, Haar Cascade, Local Binary Pattern Histogram

Table of Contents

List of Figures

List of Tables

List of Appendices

CHAPTER I
INTRODUCTION

Background of the Project

Numerous nations presently utilize public video surveillance cameras as an essential tool to screen populace occurrences, developments, and prevention of crime and terrorism, both within the private and public sectors. Facial Recognition technology is now developing and typical in a growing number of places around the globe; countries such as the US, China, India, Singapore, United Arab Emirates, and many more countries are utilizing and improving this technology. In India, facial recognition is used for crime-related incidents, mostly missing person/children's cases. The US plans to use facial recognition technology in their airports by 2021. China is currently using the technology in its schools to monitor student behaviour and record attendance. In July 2019, a South China Morning Post reported that China is currently working on the world's most advanced and powerful face recognition technology to detect and identify any of their 1.3 billion population within 3 seconds [1].

A study entitled Motion Detection and Face Recognition for CCTV Surveillance System obtained a face detection success rate of 76% and a face recognition success rate of 60%. In the said study, Haar Cascade Classifier is used for face segmentation and uses three algorithms for face extraction and training, namely Speed-Up Robust Features (SURF); Principal Component Analysis (PCA); and Counter Propagation Network (CPN). Our study uses the same Haar Cascade Classifier for face detection and Local Binary Pattern Histogram for training model and face recognition. It improves face detection and faces recognition success rate primarily because of LBPH paired with the Haar Cascade Classifier. Face Recognition-based Real-time System for Surveillance is another study with access control features to secure environments, identify individuals at a particular place, and intruder detection that uses the same algorithm for face detection and faces recognition as stated in the first study. These studies indicate that Principal Component Analysis (PCA) paired with Haar Cascade Classifier achieves acceptable face detection and recognition rate. Compared to the previously published studies, the Haar Cascade Classifier paired with LBPH shows improved results.

This project aims to fill in the surveillance camera's shortcomings. Today, universities and colleges seek technological innovation to re-open their campuses

safely for the already started school year. The resumption of face-to-face classes is a challenge for academic institutions such as WMSU. The proposed system provides a better way to monitor, identify and give better security to every student, teacher, and staff of Western Mindanao State University campus with hygienic, seamless, and preferably touchless, high-assurance identity management.

This study aims to enhance and improve the security and log management of Western Mindanao State University by integrating facial recognition technology in the school's main campus gate. This proposed system will bring new technological innovation and upgrade surveillance cameras and add new capabilities.

Statement of the Problem

Western Mindanao State University's outdated security measures are insufficient to monitor every entry to the school grounds. The present log system is still done manually using pen and paper, which adds time and paperwork to the process. Because security personnel is insufficient to protect the vast number of persons who walk in and out of a large university, such as WMSU, they must implement a facial recognition system to supplement the existing security measures. Visitors, students, and employees will find it easier and more convenient to enter the school grounds with face recognition and the ability to monitor and register each entry. This project aims to strengthen WMSU's security and upgrade the university’s present log-in system.

Objectives

This project's general objective is to develop a system that implements public security through Facial Recognition at the Western Mindanao State University Campus's main gate.

Specifically, this project sought to achieve the following objectives:

- To develop a facial recognition system for Western Mindanao State University (WMSU) that uses Haar Cascade for face detection and Local Binary Pattern Histogram for face recognition and can store and record student, faculty, staff, and other university stakeholders' personal information.
- To achieve a reliable and functional facial recognition system with a confidence rate of up to 92 percent for facial recognition.

- To develop a facial recognition system with an automatic log management capability to help Western Mindanao State University security officials distinguish between permitted and unauthorized personnel entry.
- To encourage Western Mindanao State University students and employees to use facial recognition technology to manage their logs differently.
- To prevent any crime-related incidents by keeping track on both registered and unregistered individuals.

Scope and Limitations

Scope

The developed system offers WMSU a biometric way of monitoring its students, faculty, staff, and other stakeholders using a facial recognition system. The researchers carried out the study inside the WMSU campus, with students, faculty, and staff from the College of Computing Studies serving as respondents or subjects.

The system has four essential features or functions to complete the entire facial recognition process. Face detection is a system feature that allows it to detect a subject's face, specifically facial features such as the eyes, nose, and lips; it is utilized both in registration and recognition processes. The system's generated dataset feature will be used when registering a subject. An administrator must generate a dataset to be stored in the system; the administrative personnel, in this case, are the WMSU security officials. To execute the system successfully, the administrator must be familiar with all of the system's key functions. After successfully producing a dataset in the system, the training classifier occurs. The system will train all of the subject's images to extract data in this feature. Then there's facial recognition, a system feature that recognizes a registered subject by their face and creates a log for each one. Subjects who are not registered in the system may be discovered by the system and branded as unauthorized detected.

Limitations

For the system to work, it must meet some prerequisites. Face detection may not work if the subject isn't standing directly in front of the camera and isn't more than 2 meters away. Subjects should remove all facial gear, such as helmets and masks, which will cause the facial recognition to fail. Face accessories such as spectacles and sunglasses are discouraged since they reduce the accuracy of facial recognition. All critical facial features such as the eyes, nose, and mouth must be visible in the camera feed for face detection and recognition to work. Because the system is set up at the school's entry points, it may not recognize subjects who enter the school by other channels.

Significance of the Study

This study aimed to see how effective installing a facial recognition system at the Western Mindanao State University campus gates would be. WMSU will be the first institution in Western Mindanao to use facial recognition as a biometric for log management and monitoring school housing if the project is successful. Students, parents, instructors, staff, and other university stakeholders will also benefit from the project.

These sectors will be the subjects of the system for which their face registration is mandatory to enter and exit the school grounds. This study will also benefit future researchers who study the same field. The data and idea of the project can be used as a reference in developing new systems related to face detection and facial recognition. The data presented can also be used in testing the validity of related studies and serve as their cross-reference, which will help them develop new innovative systems.

Theoretical Framework

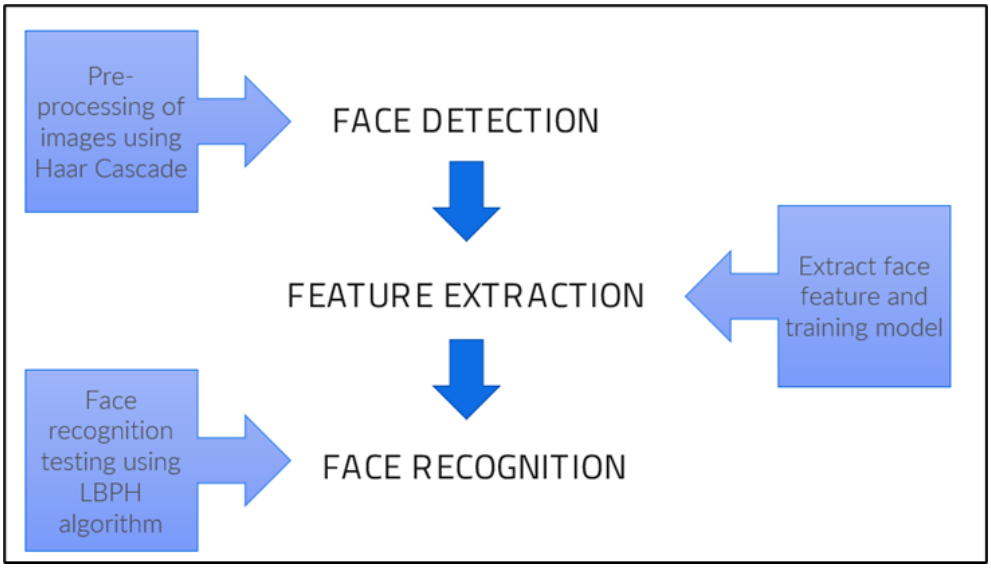


Figure 1: Theoretical Framework

Figure 1 shows the outlook of the systems processes; face detection of subjects using Haar cascade, feature extraction. The system extracts a facial feature of the subject and trains it to create numerical data. Finally, face recognition using LBPH, where the system compares subjects detected faces to the stored data to find a match for face recognition.

Definition of Terms

TERM	DEFINITION
Facial/Face Recognition	A biometric software application capable of distinctively verifying or identifying a person by analysing and comparing patterns based on the person's facial shapes is referred to as face recognition.
Face Detection	Face detection lets you find the location (pixel coordinates) of any faces in an image.

Face Extraction	A neural network takes an image of the person's face as input and outputs a vector that represents the essential features of a face.
Image Acquisition	Image Acquisition is the first stage of any vision system that refers to the collection of data required to form an image.
Training image	The training process involves finding a set of weights in the network that proves to be good, or good enough, at solving the specific problem.
Face Database	Database containing student and employee face
Face embedding	Face embedding is a vector that represents the features extracted from the face.
Confidence rate	Refers to a scale that describes how certain a facial recognition system that a match it has produced is accurate.
Haar Cascade	This refers to an Object Detection Algorithm used to identify faces in an image or a real-time video. The algorithm uses edge or line detection features proposed by Viola and Jones.

Local Binary Pattern Histogram (LBPH)	This refers to a face recognition algorithm based on a local binary operator designed to recognize both the side and front face of a human.

Table 1: Operational Definition

CHAPTER II
REVIEW OF RELATED LITERATURE

The review of related literature is created to summarize existing research or studies related to this project. This chapter contains the related literature and related studies.

Related Literature

CCTV

CCTVs were first invented and used in Germany. It was a system developed to monitor the country's V-2 rockets. Until people realized its usefulness, it was launched commercially in 1949 for the public. Years have passed since it was invented, and CCTVs evolved into state-of-the-art equipment with several applications in many businesses and homes during this period.

According to Koorsen Fire & Security, CCTV is everywhere these days. You'll find them in pretty much every bank, school, hospital, and business of every size, even in many homes. Wherever you find them, they act as an extra eye whenever people are not around. They are there to prevent crime, monitor the environment, and, most importantly, record video footage that may be important. What does CCTV mean, and how does it work? Closed Circuit Television (CCTV) or simply Video Surveillance Systems are unique cameras that people use to monitor security and other reasons. For CCTV to work, it must have these four parts: cameras, recording devices, monitors, and video management software [2].

Schools these days are also implementing the use of CCTV to benefit from its usefulness, whether in campus security, attendance system, automated registration, and emotion detection.

Facial Recognition

According to a study by Parekh Payal and Mahesh M. Goyani, Face Recognition identifies and verifies faces. Face recognition has vast importance in Security, Healthcare, Banking, Criminal Identification, Payment, and Advertising. The

study determines various challenges and techniques for face recognition. Challenges such as illumination, pose variation, facial expressions, occlusions, aging, etc., are the critical challenges to face recognition. The methods used for facial recognition to work are pre-processing, Face Detection, Feature Extraction, Optimal Feature Selection, and Classification are the primary steps in any face recognition system. The study also states that facial extraction is the most crucial stage for any facial recognition system to work and the deep learning method helps the user by freeing them from handcrafting the features [3].

Another study by Sign Modou Bah and Fang Ming states that much development has been made in facial recognition and Detection for security, identification, and attendance purposes. Still, it turns out that there are still issues that hinder progress from reaching or surpassing human-level accuracy. These issues are varieties in human facial appearance such as changing lighting conditions, clamor in confronting pictures, scale, posture, etc. With these challenges to consider, the researchers of this study used another method by using the Local Binary Pattern (LBP) algorithm combined with advanced image processing techniques such as Contrast Adjustment, Bilateral Filter, Histogram Equalization, and Image Blending. These solve some of the issues hampering face recognition accuracy to improve the LBP codes, thus improving the accuracy of the overall face recognition system [4].

Neural Network

In a research done by Naoyuki Kubota titled "Applying Artificial Neural Networks for Face Recognition," in the process of identifying or analyzing the facial land points, the researcher used a hybrid model combining AdaBoost and Artificial Neural Network (ABANN) to improve the efficiency of facial recognition; the system consists of four modules: Detection, alignment, feature extraction, and matching; The research was done by Naoyuki Kabuto provides some basic neural network models and efficiently applies these models in modules of the face recognition system. For the face detection module, a three-layer feedforward artificial neural network with Tanh activation function is proposed that combines AdaBoost to detect human faces. The face detecting rate is rather high [5].

The Naoyuki Kubota facial recognition with neural network uses three-layer feedforward artificial neural network and AdaBoost for detecting human faces, which, in comparison with traditional technique, shows a higher identifying and detecting rate.

While in a research done by Juan Pablo Ballerini, Sergio Nesmachnow, and Martín Rodríguez of Universidad de la República, Facultad de Ingeniería titled "Facial Recognition Using Neural Networks over GPGPU" they introduce the use of parallel neural network approach which also use the processing power of Graphics Processing Unit (GPU); in their studies, the three researchers thought that the efficiency of Neural network processing image in detecting facial points would be effective and efficient with the help of GPU which they stated that "GPU implementations allow obtaining a significant reduction in the execution times of complex problems when compared with traditional sequential implementations on CPU" and with the use of sequential algorithm by Shufelt and Mitchell for recognizing if a given picture is of a particular person [6].

The method of Naoyuki Kubota slightly improve the analyzing process with the use of a hybrid model and combing AdaBoost and Artificial Neural Network for detecting and matching face, but in the test experiments in proving that AdaBoost and ANN approaches for detecting faces do not achieve good results of performance time and detecting rate yet.

While the research done by Juan Pablo Balarini, Sergio Nesmachnow, and Martín Rodríguez of Universidad de la República has a much better outcome with a significant reduction in computing times can be obtained, allowing solving large instances in a reasonable time. Speedup greater than eight is achieved when contrasted with a sequential implementation, and a classification rate superior to 85 % is also obtained.

Deep Learning

Deep Learning is a subfield of Machine Learning which uses a 'layered' architecture to learn representations. Each successive layer in the layered architecture works with more meaningful and sophisticated representations; in a research done by Diego Andina (2017) titled "Deep Learning for Computer Vision," he talks about how deep learning evolves in a short period according to Diego Andina "Deep learning methods have been shown to outperform previous state-of-the-art machine learning techniques in several fields, with computer vision being one of the most prominent cases" which is correct from unlocking your smartphone to facebook improve tagging system; deep learning evolve on how the computer interprets and analyzing data, in

his research he stated that Deep learning allows computational models of multiple processing layers to learn and represent data with multiple levels of abstraction mimicking how the brain perceives and understands multimodal information [7].

The researcher Diego Andina also indicates that there are 4 most significant deep learning schemes used in computer vision problems, that is, Convolutional Neural Networks, Deep Boltzmann Machines and Deep Belief Networks, and Stacked Denoising Autoencoders, which can be used for systems like object detection, face recognition, action and activity recognition, and human pose estimation.

In a research done by Md Nazmus Saadat and Muhammad Shuaib (2020) titled "Advancements in Deep Learning Theory and Applications: Perspective in 2020 and beyond, "the researcher talks about how deep learning works; the researcher stated that there are two types of machine learning; first is the supervised learning, the machine can only give you correct output when the input is already experienced in training phase; it is based on experience; the more is the training dataset or experience of your machine the higher are the chances of getting the actual output, and the second is unsupervised learning, supervision of a model is not needed, instead the model work on its catches new data and discovers the information inside the data. It usually deals with label-less data; compared to supervised learning, unsupervised learning is more complicated. It is usually used to find features and unknown patterns [8].

Foreign Studies

Motion Detection and Face Recognition for CCTV Surveillance System

According to this study, the use of CCTV evolved from simple surveillance of a home, workplace, and private properties into an intelligent control device. This study CCTV video processing has three outputs: face detection information, motion detection information, and face identification information. Accumulative Difference Image or ADI was used for motion detection, and Haar Classifiers Cascade was used for facial segmentation. To extract features, this study made use of Speedup Robust Features or SURF and Principal Component Analysis or PCA. Counter-Propagation Network or CPN is used to educate the features. The study used 45 CCTV videos and all 45 videos were subjected to offline checks. The study's test results show a 92% success rate for motion detection, 76% for face detection, and 60% for face identification. According to the findings, the study concludes that using CCTV video with a natural context for face

detection did not achieve optimum resolution, while the motion detection method is preferable to use in real-time situations [9].

Face Recognition-based Real-time System for Surveillance

According to this research paper, the ability to automatically recognize human faces based on dynamic facial images is essential in security, surveillance, and the health/independent living domains. Specific applications include access control to secure environments, identifying individuals at a particular place, and intruder detection. This research proposes a real-time system for surveillance using cameras. The process is broken into two steps: (1) face detection and (2) face recognition to identify persons. For the first step, the system tracks and selects the faces of the detected persons. An efficient recognition algorithm is then used to recognize detected faces with a known database. The proposed approach exploits the Viola-Jones method for face detection, the Kanade-Lucas-Tomasi algorithm as a feature tracker, and Principal Component Analysis (PCA) for face recognition. This system can be implemented in different restricted areas, such as at the office or house of a suspicious person or the entrance of a sensitive installation. The system works almost perfectly under reasonable lighting conditions and image depths [10].

This research paper is related to our study for it tackled the use of CCTV in implementing facial recognition systems at different restricted facilities, such as workplaces, houses, and schools. The study also aims to use the system for crime monitoring and prevention and Detection of suspicious individuals, which is the same objective that our research has.

Local Studies

Student Monitoring System of Our Lady of Fatima University Using Face Recognition

This research paper concludes that the rapid development of face recognition is due to a combination of factors: active development of algorithms, the availability of large databases of facial images, and a method for evaluating the performance of face recognition algorithms. The system covers any departments, agencies, or companies which require personal identification and security for their employees. The face recognition system covers multiple face photos, matching faces, head rotations, and detects 66 facial feature points (eyes, eyebrows, mouth, and nose), and all data are

placed in a database. Additional enrolments will be required upon various changes in registered faces. The said system only limits personal identification, which contains specific fields about the registered user, and it cannot detect the skin color and age of a person, and the system is not a video-based face recognition system. This system does not expect to solve all the issues in face recognition, such as extreme facial expression, wearing on the face, excellent age discrepancy, extreme lightning conditions and without frontal face information [11].

Synthesis

Historically, security systems at Western Mindanao State University (WMSU) have consisted of a handful of critical components. These include well-trained personnel and a CCTV system. Occasionally, the typical security system components are sufficient for specific purposes. Nonetheless, there is no denying that these security systems have inherent flaws that more advanced and sophisticated systems are more than capable of overcoming.

Recently, the campus integrated a new method of log and identification in the school gates using Radio Frequency Identification (RFID) in students/employee ID, though this might be a good solution. It takes an additional step when entering the school premises, through the development of Face Recognition for more enhance security and better log management system this is set to overcome the weakness of the traditional security system of the campus.

Various face attendance systems also already exist like the "Motion Detection and Face Recognition for CCTV Surveillance System" by Nurhopipah & Harjoko, 2018, their study shows promising results in detecting motion in surveillance cameras, reaching the confidence rate of 92%. However, the face recognition part of the system only reaches up to 60%, similarly Alolor along with other researchers also developed a system called "Student Monitoring System of Our Lady of Fatima University Using Face Recognition" the system where able detects 66 facial feature points (eyes, eyebrows, mouth, and nose) and all data are placed in a database. However, their system is not a video-based face recognition system.

The three related studies mentioned above utilized the same algorithm for face detection, the Haar cascade algorithm. For face detection, the studies used a different approach, which gives different results regarding the effectiveness and accuracy of

face recognition. With this, the researchers of this study decided to use the Haar cascade for face detection. They paired it with another face recognition algorithm that was not used by the mentioned related studies. This study's algorithm used for face detection was the Local Binary Pattern Histogram (LBPH). The researchers have found that using much higher pixel size training images of each Subject and with much better parameter values of the haar cascade algorithm paired with LBPH results in much higher accuracy and stability in developing the system.

Comparison Table of Related Studies

System	Image acquisition from video frame	Face Detection	Face Recognition	System Alarm for recognized and unrecognized individuals	Automatic Log feature	Face Registration	Video log history	Record log with date and time
NHANCED SECURITY ND LOG SYSTEM FOR VESTERN MINDANAO STATE UNIVERSITY VMSU) USING A REAL- IE FACE RECOGNITION SYSTEM	✓	✓	✓	✓	✓	✓	✓	✓
Face Recognition Based Automated Student Attendance System	✓	✓	✓		✓	✓		
FacePro: Panasonic Facial Recognition System	✓	✓	✓	✓		✓		
Face Recognition Security System	✓	✓	✓	✓		✓		

Applock (Android App)		✓	✓					

Table 2: Comparison table among similar published system

The table above shows the comparison of features of the systems. This is the list of systems that are similar to our system in terms of the primary purpose of facial recognition and detection. As shown in the table, the facial recognition system for WMSU dominated other existing systems in terms of features. With the eight features listed, the study's system can perform all the features.

CHAPTER III
METHODOLOGY

Research Design

This study mainly provides Western Mindanao State University with a face recognition biometric for log entry records on the school premises. The face of the subject is used as a dataset and utilized using the Haar cascade algorithm for face detection and the Local Binary Pattern Histogram for face recognition.

The study will use the applied research design method to determine the applicability and effectiveness of Haar Cascade and Local Binary Pattern Histogram to enhance school security monitoring and log record management. During the development process, the researchers will follow the iterative model for the software development:

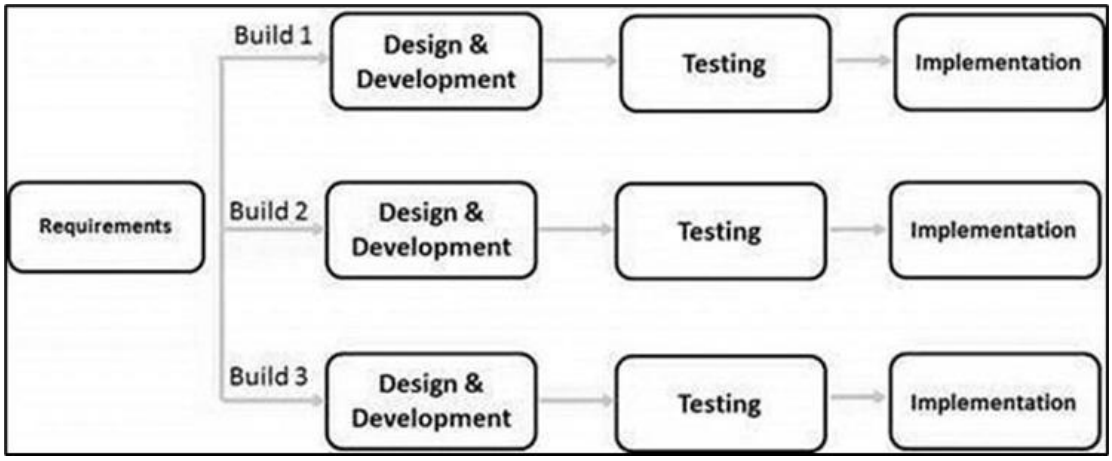


Figure 2: SDLC Iterative Model

Phase #1: Planning and Requirements: In this stage, the researcher will map out the initial requirements, gather the related documents related to face recognition and face recognition algorithm, and create a plan and timeline for the first iterative cycle.

Phase #2: Analysis and Design: in this phase, the researcher will finalize the requirements needed, the database models, and technical requirements based on the

plan. Then create working architecture, schematic, or algorithm and install all the required library and development tool that satisfies the requirements.

Phase #3: Implementation: The researcher will start developing the functionality and design required to meet the system's specifications.

Phase #4: Testing: the researcher will identify and locate what's not working or performing to expectations.

Phase #5: Evaluation and Review: Compare this iteration with the requirements and expectations.

After completing these steps or processes, it's time to tackle the next cycle. In this process, the product goes back to step one to build on what's working. Identify what the researcher learned from the previous iteration. This iterative development sometimes called circular or evolutionary development, is based on refining the first version through subsequent cycles, especially as you gather and include requirements. It allows the developer to remain flexible as the researcher identifies new needs or unexpected system issues.

Respondents

This study requires several respondents to test the system's effectiveness in terms of implementation. The respondents of this study must be WMSU employees, faculty staff, and students, for they will be the main subject of this research. To minimize the population of WMSU, the researchers select the respondents by picking one college department from the cluster of colleges in WMSU, the College of Computing Studies (CCS). This study used a simplified random sampling method. These randomly selected respondents consist of faculty members and students of CCS.

Research Instrument

The study aimed to determine the effectiveness of facial recognition as a biometric to record logs of entry on the school campus using Haar Cascade paired with Local Binary Pattern Histogram algorithms.

The researchers gathered data that needed to be inputted into the system, like the individual's basic information (Names, Student ID, Course, Types) and their facial features. The face is the primary data for this study. Face features are then converted into a histogram to store them in the system.

The first procedure to acquire and prepare the data sets was by portrait acquisition; during this method, the system must first sight the presence of the face within the webcam. An individual only needs one (1) portrait to be used as a data set. After detection and capturing the portrait, it will proceed to the pre-processing procedure to get a BGR image and cropped faces of equal-sized pictures.

Data Gathering Instruments, Techniques, and Procedures

Image Acquisition

The Face Recognition system obtained 200 images per person by interfacing with a webcam.

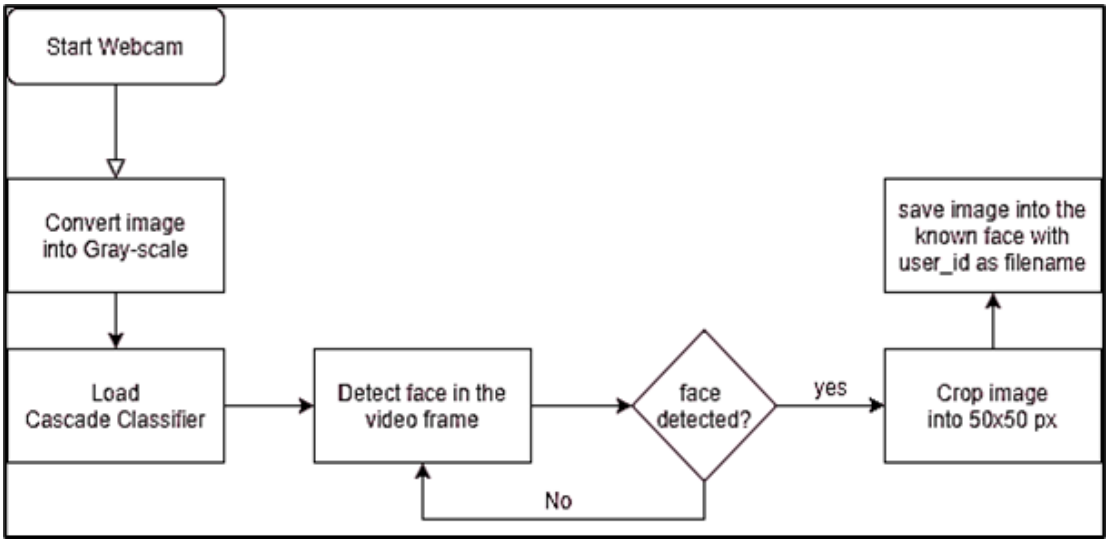


Figure 3: Image Acquisition Flow Chart

The above flowchart shows the data gathering procedure during the image acquisition phase; this allows the researchers to prepare the datasets for the following process.

Datasets Acquisition

A data set was created by taking images from a camera or images already saved and then creating a unique identifier in each image, such as name, age, profession, etc. Images with unique identifiers will then be transferred to the database. It is recommended to take many sample images from a single person for these samples will be used for training algorithm and testing.

Algorithm

In this study, the researchers used two algorithms, namely Haar Cascade for face detection and Local Binary Pattern Histogram for face recognition. These algorithms are utilized to detect faces and collect face data and convert it into a histogram to store in the system. Every registered face in the system will have a name, age, course, college, birth date, and sex. The researchers selected these two algorithms as these algorithms provide the best results in terms of detection speed and accuracy and facial recognition speed and confidence rate.

Haar Cascade Classifier for Face Detection

This study used the Haar cascade for the detection of faces. Haar Cascade is an object detection algorithm designed to identify faces in an image or a real-time video. This algorithm is viral and is utilized worldwide. Haar cascade is used in mobile applications and surveillance security and biometrics. The algorithm is trained by first giving it positive images, consisting of faces of different individuals, and negative images that do not consist of any face. The algorithm will differentiate and learn which face is and which is not in this method.

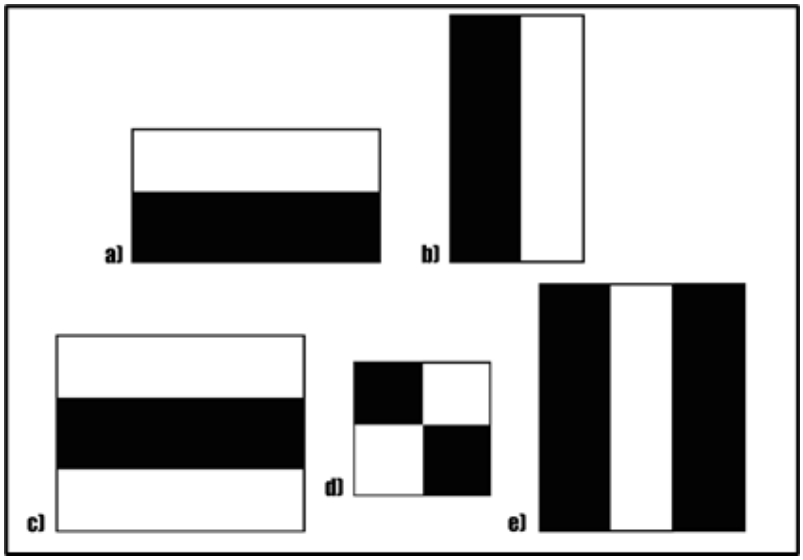


Figure 4: Haar-like Feature

Haar cascade works by using the figures above. These features are used in a single image to locate a particular feature of the face like the eyes, nose line, lips, etc. These features are necessary to collect data from the image; these data are gathered from every image pixel using these features. When these features are implemented in an image, it will sum up to approximately 180,000 features. Some of these features are irrelevant or not necessary examples are the hair and the foreheads. Haar cascade only needs critical facial features such as eyes and nose lines. Of the 180,000 features, most features are unnecessary. Therefore, AdaBoost is used to eliminate unnecessary features.

Haar cascade is one of the oldest but most tested and most powerful facial detection algorithms. This study uses the algorithm because it is a tested algorithm that proves its effectiveness for a long time. Not only facial feature detection but it is also used in plate number detection, etc. Haar cascade is also compatible with the Local Binary Pattern Histogram, making an acceptable confidence rate.

Face Recognition Local Binary Pattern Histogram

LBPH or Local Binary Pattern Histogram is a face recognition algorithm based on a local binary operator designed to recognize the front and side of the human face. Facial recognition is a technology used to identify a person from a video or image and link to a particular person. Facial recognition is different from facial detection. Facial detection is the ability to recognize human faces from video or an image, therefore used for facial recognition.

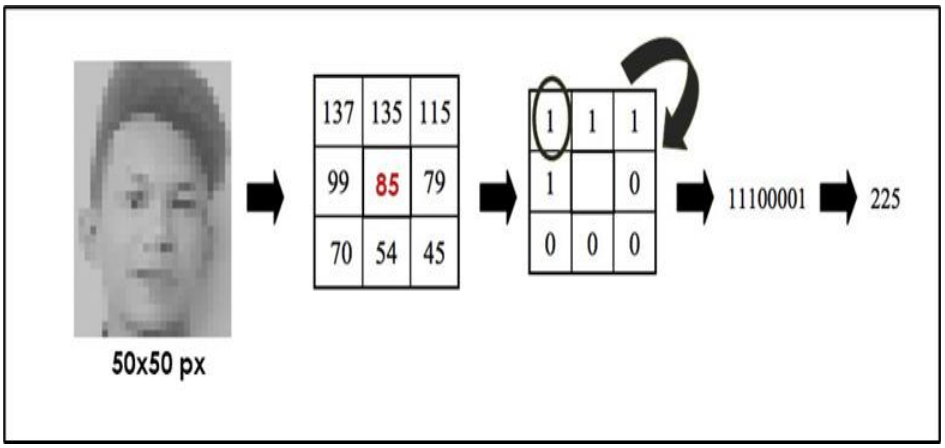


Figure 5: Image Pixel Conversion to Binary Number

Local Binary Pattern works by using three by three blocks of a pixel in an image, and each pixel has a value ranging from 0 to 225.

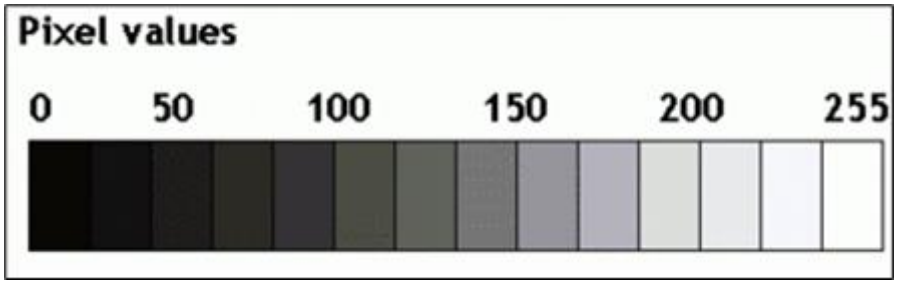


Figure 6: Pixel Value

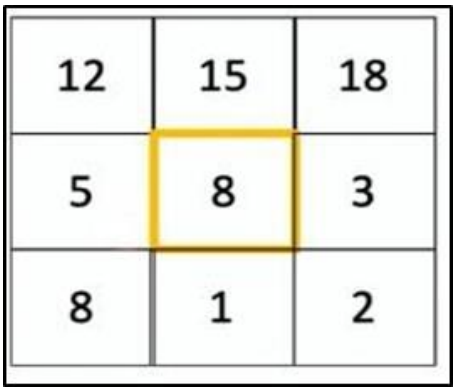


Figure 7: Image block with pixel value

Figure 7 shows a block of an image with three rows and three columns for 9 pixels. The central value of this block is called the threshold, and the surrounding values are its Neighbor. To get the necessary calculations for the LBP, the researcher compares the threshold to its neighbors with the condition: If the threshold is greater than or equal to its Neighbor, the Neighbor's value will be changed to 1; hence it will adjust zero.

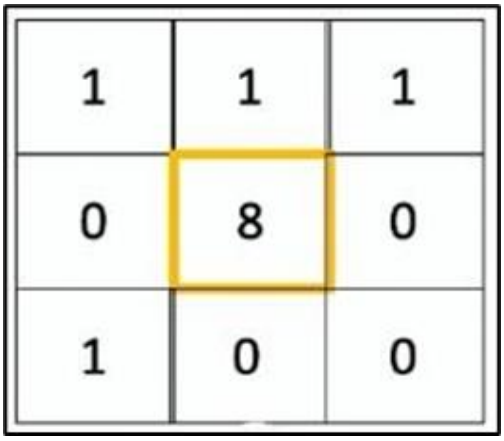


Figure 8: Image block with pixel values converted to binary numbers

This calculation will then be changed into a binary number, excluding the threshold starting from the top left corner element followed by the element of the second row creating a circular pattern up to its last element. As seen in Figure 8, it will give a binary value of 11100010.

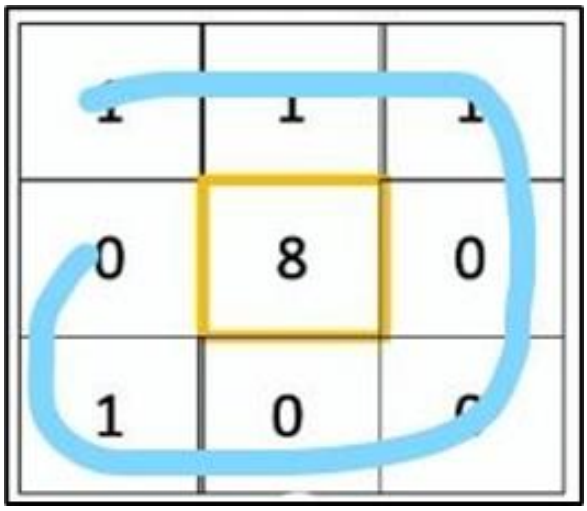


Figure 9: Image block with pixel values converted to binary numbers and grouped

The binary value will be converted into a decimal value, which indicates that the three by three block of pixels is around the central value of 266.

The binary value is converted into a decimal value, which indicates that the three-by-three block of pixels is around the central value of 266.

Output Value of LBP Operator

$$LBP = \sum_{n=0} S(in - ic)\, 2^n$$

$$S(z) = 1 \text{ if } z \geq 0$$

$$0 \text{ if } z > 0$$

Where, in = Center pixel value

Ic = Neigbour pixel values

The above formula is the proper formula for getting the binary value of the pixels surrounding the center pixel. If all neighbor pixels are converted into binary, they will be computed to get the decimal value for the LBP operator.

The last step in the LBPH is to create a histogram that indicates how many times a color appears in each square. It is how the algorithm identifies edges, borders, and corners of an image. With this process, the algorithm knows which histogram indicates the color of the eyes, the lips, the mouth, etc. The binary value will be

converted into a decimal value, suggesting that the three-by-three block of pixels is around the central value of 266.

Training the Algorithm

Figure 10: Training Images Captured with Haar Cascade

To achieve the objective confidence rate, the system required to generate 250 training images using a Haar cascade frontal face. The training images must be 300x300 pixels and reduce the RGB to grayscale images. It also needs to set an ID (it may be a number or the person's name) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID.

LBPH Custom Parameters

- Minimum Neighbor - 10
- Scale Factor = 1.2

Minimum Neighbor It is a threshold value that specifies how many neighbors each rectangle should have for it to be marked as a true positive. In other words, let's assume that each iteration marks specific rectangles (i.e., classifies a part of the image as a face). If the subsequent iterations also mark the same areas as a positive, it increases the possibility of that rectangle area being a true positive. If a particular area is identified as a face in one iteration but not in another, they are marked as false positives. In other words, minNeighbors is the minimum number of times a region must be determined as a face. In contrast, Scale Factor means the smaller the value of the scale factor, the greater the accuracy and higher the computation expenses.

Entity Relation Diagram (ERD)

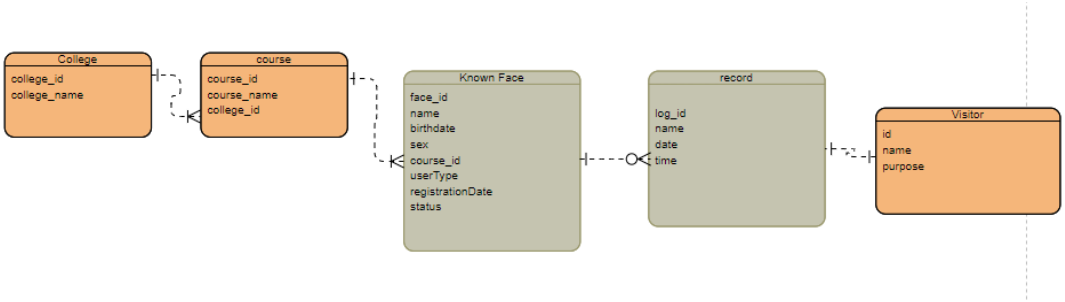


Figure 11: Entity Relation Diagram

System Use case Diagram

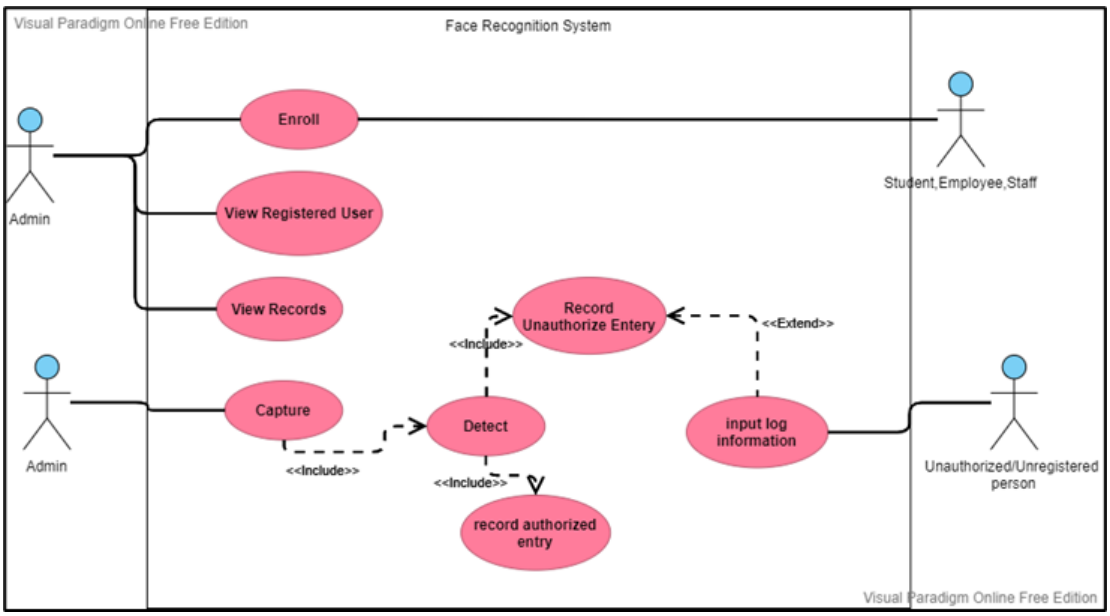


Figure 12: System Usecase Diagram

The figure above shows the steps of the transaction using the system. The subjects (students, employees, staff) will enroll their faces in the system and store the date for each Subject; the system will then recognize their entry and create a log stating the time and date of entry. The system will also recognize unenrolled subjects as unregistered individuals.

System Architecture

The system consists of a face detection module that detects the face(s) in each frame and a face recognition module that identifies detected face images from the detection module as a name or identification number. The overall architecture of a Complete Face Recognition System is shown in the following figure.

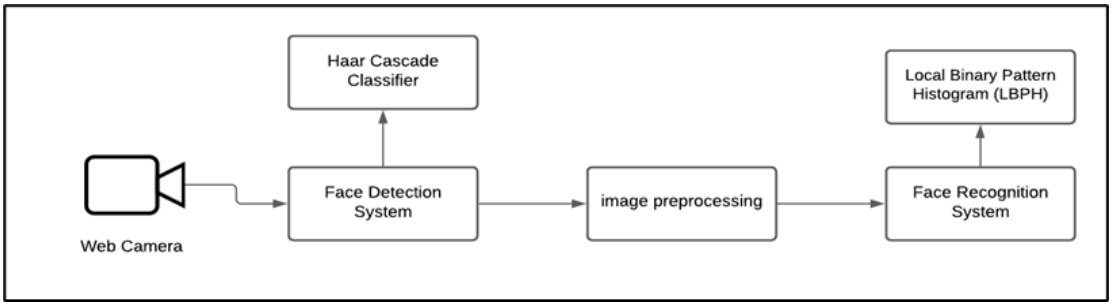


Figure 13: System Architecture

System Block Diagram

System Block Diagram, i.e., Figure 14 below served as the framework of the system. The Camera module is the main component of the system. The camera module will capture image frames when the system is turned on. If the face is detected, it would crop the face, pre-process the image using the OpenCV library, and extract the face feature using LBPH. Then it would output whether the detected face is registered or unregistered.

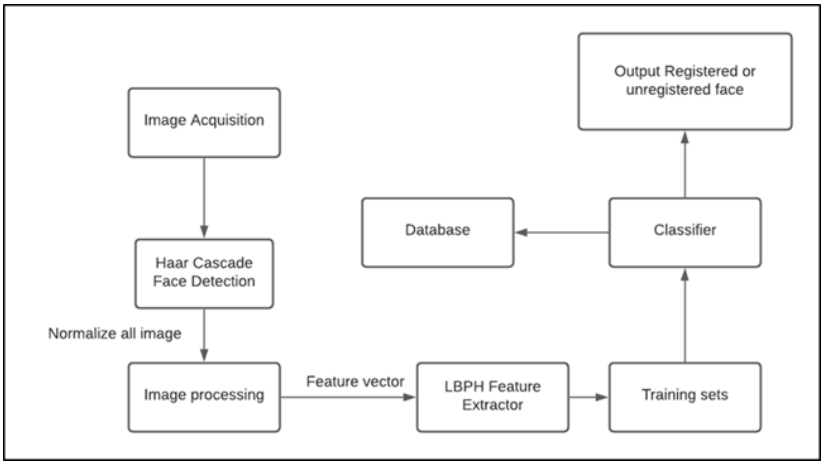


Figure 14: System Block Diagram

Hardware during development

CAMERA: Redragon HITMAN webcam

CPU: Ryzen 5 5500

RAM: 8GB

Storage : 1TB SSD

System Requirements

OPERATING SYSTEM: Windows 7 (64-bit) version later.

CAMERA : 1080p CCTV or Web Camera

CPU: Intel or AMD processor with 64-bit support, 2 GHz, or faster processor.

RAM: at least 4GB

Storage : 1TB SSD

Development Tools

IDE

- Visual Studio Code
- PyQT5 Designer

Programming Language

- Python
- PHP

Local Web Server

- XAMMP - is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

Libraries for Development

- cycler==0.10.0
- kiwisolver==1.3.2
- matplotlib==3.4.3
- mysql-connector==2.2.9
- numpy==1.21.2
- opencv-contrib-python==4.5.3.56
- opencv-python==4.5.3.56
- pandas==1.3.3

- Pillow==8.3.2
- pyparsing==2.4.7
- PyQt5==5.15.4
- PyQt5-Qt5==5.15.2
- PyQt5-sip==12.9.0
- PySide2==5.15.2
- python-dateutil==2.8.2
- pytz==2021.3

Network Requirements

The Client-side must be connected on the same network as the server.

CHAPTER IV
RESULTS AND DISCUSSION

The proposed Thesis, entitled "Enhanced Security Monitoring and Log System for Western Mindanao State University using Real-time Face Recognition System," uses the Local Binary Pattern Histogram (LBPH) algorithm for face recognition and Haar Cascade for face detection over a Red Dragon HD web camera. The system operates by following these processes.

First, obtaining a dataset of facial images, second, training the dataset, and third performing facial recognition. In the facial image dataset acquisition process, a subject's face is scanned using the camera to register the Subject's facial dataset in the system. Every Subject has captured 250 images upon registering their facial image in the system. The subjects are assigned a unique ID and their information during this process, such as name, course, etc. These 250 images of every Subject then go through a dataset training process, where the LBPH algorithm is used to create a unique histogram of every registered Subject. After a successful Face Registration to the system, the newly registered Subject were scanned again for face recognition. If the Subject's face has been successfully recognized by the system ethe process of registering and face recognition ends.

Face Enrolment

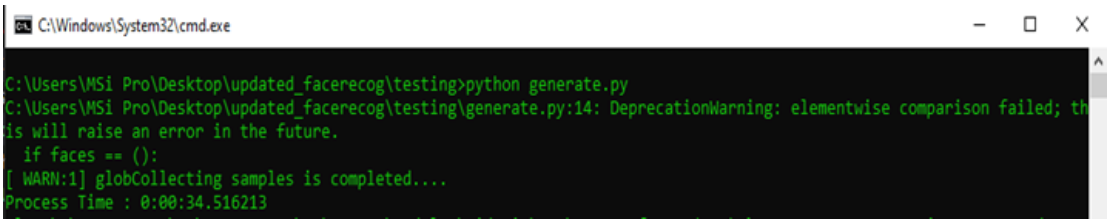


Figure 15: Face Enrolment Real-time Generation Speed

Face Enrolment is the first process of the system. A web camera module is used to detect the face of the Subject using the Haar cascade algorithm and then capture 250 images per Subject. Upon testing, it took 34.616213 seconds to generate 250 images of a subject for registration.

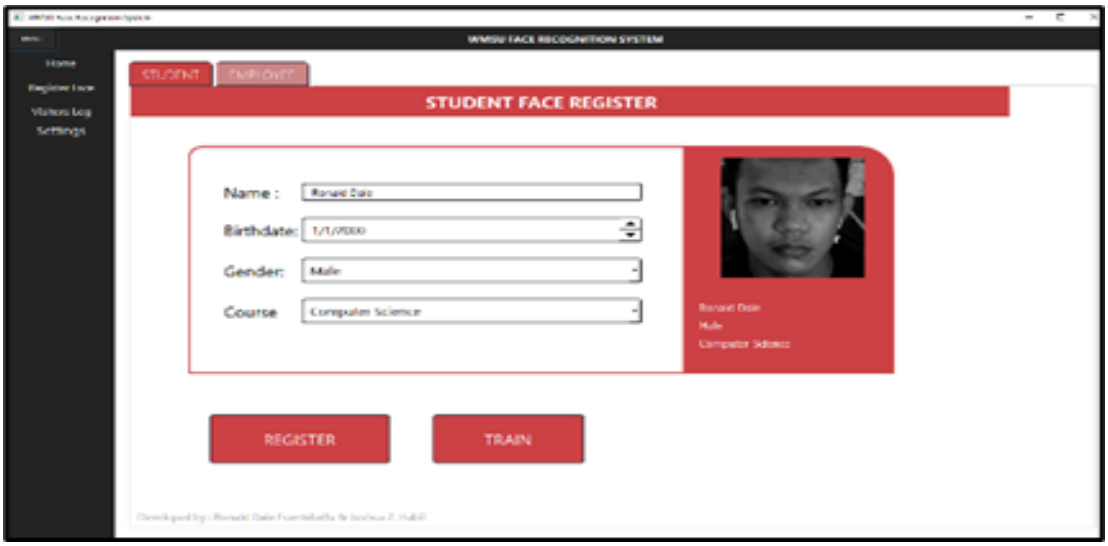


Figure 16: Face Registration UI

Face Enrolment for students requires basic information like name, birth date, gender, and course; for employees, the registration information requires name, birthdate, gender, department, employee type, and position. During the enrolment process, the system then assigns the Subject a face ID to be able to identify the Subject's face uniquely.

This test proves that the system can store and record facial images with the personal information of each Subject to be used for face recognition.

Training the Classifier/Dataset Training

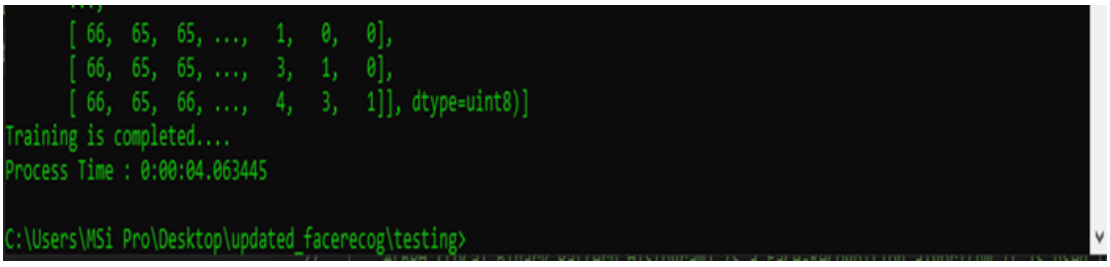


Figure 17: Training time with only one registered subject

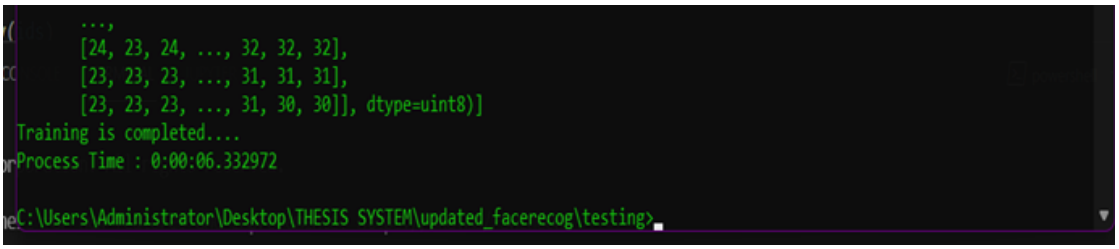


Figure 18: Training time with two registered subjects

Figure 19: Training time with three registered subjects

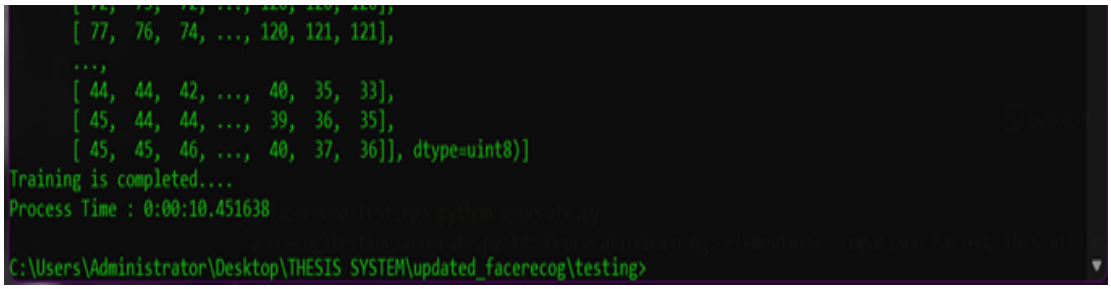


Figure 20: Training time with four registered subjects

Training classifiers or dataset training is a process where the captured 250 images are trained to create a unique identification per Subject. The ID of each unique Subject is used for the face recognition process.

Training Classifier			
No. Image	**ID**	**Process Time**	**Classifier File Size**
250 Image	Subject ID 1	4.06 seconds	9.87 MB
250 Image	Subject ID 2	6.33 seconds	19.74 MB
250 Image	Subject ID 3	8.98 seconds	29.61 MB
250 Image	Subject ID 4	10.45 seconds	39.48 MB

Table 3: Training Classifiers

The table above shows that the first registered Subject took only 4.06 seconds to train the classifier. However, the second registered Subject took 6.33 seconds. Based on gathered data, the processing time of each new registration increases by 2 seconds on average. The system generates a classifier that increases its size by 9.87 MB each time a new Subject is registered.

This test data shows that the system's model is fast and lightweight. It requires 250 images of each Subject for the best result.

CPU and Memory Consumption

When idle or without activating the face recognition, the system only uses 0.26% of CPU and 100.3MB of Random Access Memory (RAM). However, after launching the face recognition with only one (1) webcam enabled, the CPU usage was 23.7.0% on average, 25.0% at maximum, and the average RAM consumption was 155.5 MB. After Enabling face recognition with two (2) cameras, the CPU is still at 25.0%. On average, it's only 19.0%, but the RAM usage increases up to 340.9MB, a 185.4MB difference between the first test with only the first camera and the second test with two cameras enabled. The testing and observation were done using System

Gauge Software on Ryzen 5 5000 Series CPU with 8 gigabytes of RAM on Windows 10 Operating System.

Figure 21: Data while using only 1 camera for face recognition

Figure 22: Data while using 2 web camera for face recognition

Testing Experimentation

The system creates the ID of each successfully registered Subject with their personal information such as name and course when producing the dataset. The study used five subjects and registered their faces to the system for the testing. In the testing period, the researcher checks the speed and accuracy of the system. and lets the subjects pass through the web camera to gather data on how fast the system can detect and recognize a moving subject.

Based on the testing results, the system can detect and recognize a subject walking at an average speed and walking in hurry subjects running over the web camera are failed to be detected. These results show that the system is ideal for

WMSU's security check, where individuals who enter are required to stop for a check-up.

The expected results of facial recognition testing are defined by these terms below.

Figure 23: Registered face used for testing the system

True Positive

True positive occurred when a detected, and recognized Subject matches the available one in the systems stored training dataset. The system will then show the information of that recognized Subject; in this scenario, the result is accurate.

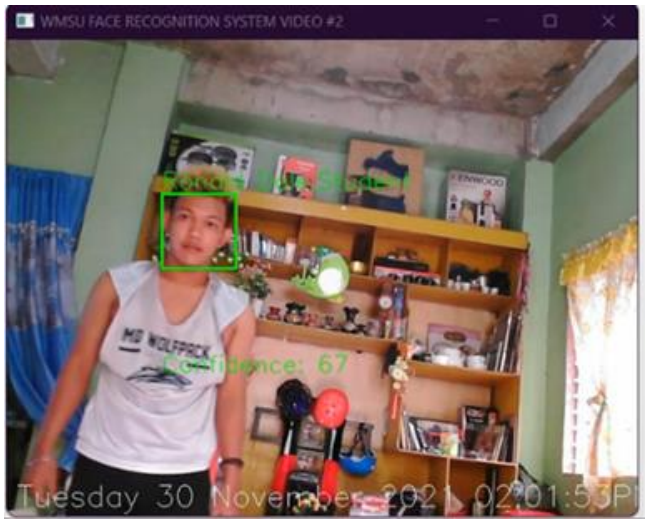


Figure 24: Sample of true positive occurrence

True Negative

The true negative occurs when a subject's dataset is not stored in the system's database, and the system cannot recognize the Subject. The system will then show an "UNAUTHORIZED" notification that this result is accurate.

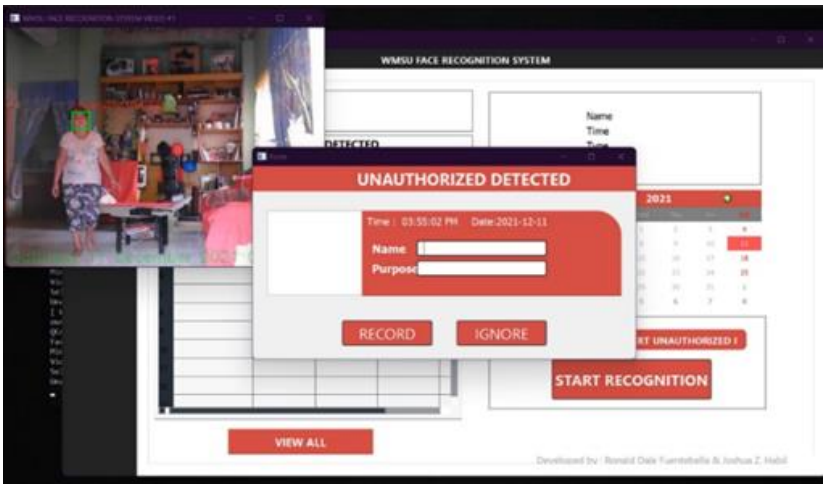


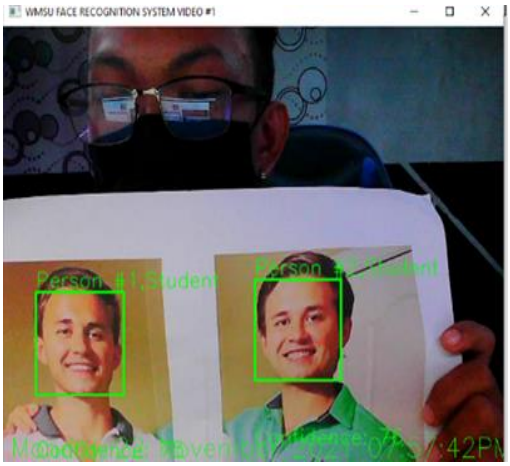
Figure 25: Sample of true negative occurrence

True Occlusion

True occlusion happens when a subject's dataset is without occlusion. Occlusion in facial recognition means objects that might make facial recognition fail, like wearing masks or helmets (ex. shades, glasses, etc.) saved in the system's database. The recognized Subject matches the one in the training dataset. The system will display the Subject's information, which is accurate. (Refer to Testing the System with Change of Appearance.)

Figure 26: Sample of true occlusion occurrence

Testing the System with Twins



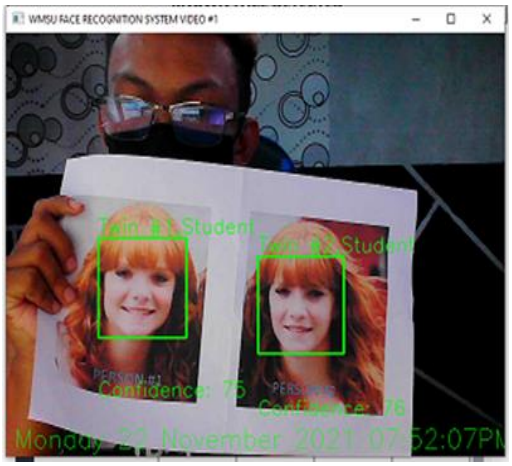


Figure 28: First twin sample testing

Figure 27: Second twin sample testing

Figure 27: Twin Test 1		
Subject	Confidence Rate	Result
Person 1	77	True Postive
Person 2	76	True Postive

Table 4: Twin 1 test results

Figure 28: Twin Test 2		
Subject	Confidence Rate	Result
Twin 1	76	True Postive
Twin 2	77	True Postive

Table 5: twin 2 test results

The two figures above showed testing of the system with twins or subjects with similar facial looks as observed from the testing. The system can recognize which is which from these subjects since it uses the LBPH algorithm, which converts every Subject's facial dataset into a unique histogram for facial recognition.

This test gives the system reasonable assurance and reliability, which shows that the system can still recognize and differentiate individuals with similar facial features and appearance, such as twins.

Testing the System with Change of Appearance/Occlusion

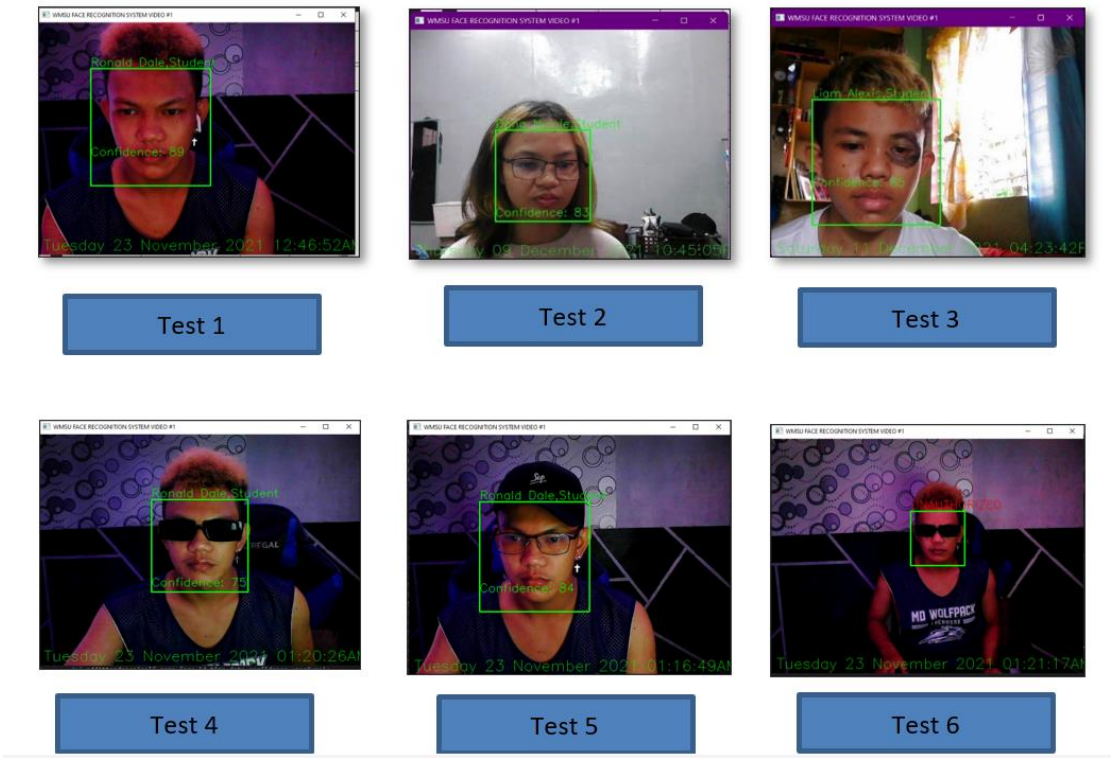


Figure 29: Face detection and recognition with change of appearance/occlusion

Test Number	Change in Appearance	Lighting Condition	Confidence Level	Occlusion
1	No Change	Artificial light	89%	No occlusion
2	black eye makeup	Artificial light	85%	makeup
3	With eyeglasses	Artificial light	83%	Eyeglasses
4	With eyeglasses and a cap	Artificial light	84%	Eyeglasses and cap

5	With shades	Artificial light	75%	shades
6	With shades but a bit far from the camera	Artificial light	Undetectable	Shades but with distance from the camera

Table 6: Analyzing data with face occlusion testing

The testing results showed that the system can still recognize individuals with minor changes in appearance, such as having a beard, blackened eye, or wearing eyeglasses and a cap, with a high confidence score of up to 89%. However, on test #5, where the Subject is wearing a sunglass, the confidence level dropped to 75%. With some minor changes in lighting conditions, the system already failed to recognize the Subject.

This test demonstrates that the system is sufficiently adaptable and reliable to recognize faces with subtle changes in appearance. This data is advantageous since many people use facial accessories such as eyeglasses, enabling the system to identify individuals more efficiently.

Testing the System with Pose Variation

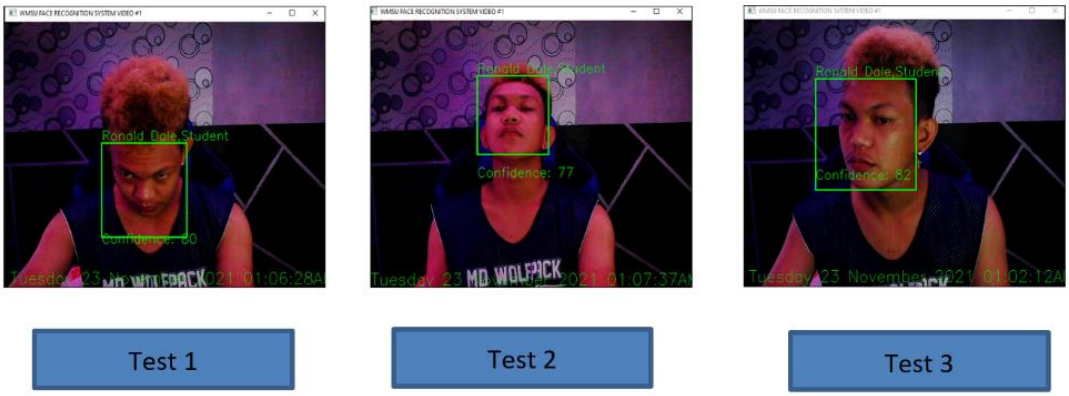


Figure 30: Testing with pose variation

Test	Pose Variation	Status
1	Head is tilted downwards at 30 degrees	Recognized
2	Head is tilted upwards at 30 degrees	Recognized
3	Head is tilted sideward at 25 - 30 degrees	Recognized

Table 7: Pose variation analysis

Three poses, as shown above, are used to test if the system can still detect and recognize the Subject. Pose variation happens when the Subject's position is at a different angle from the camera. The system can still detect and recognize the Subject on three different pose angles, but with a different confidence level depending on the angle of the face.

This test was conducted to guide the angle and place the camera. Test results showed that even though the Haar Cascade Classifier is used for frontal-face Detection, the system can still detect and recognize faces tilted downward, upwards, and sideward up to 30 degrees.

Confidence Test

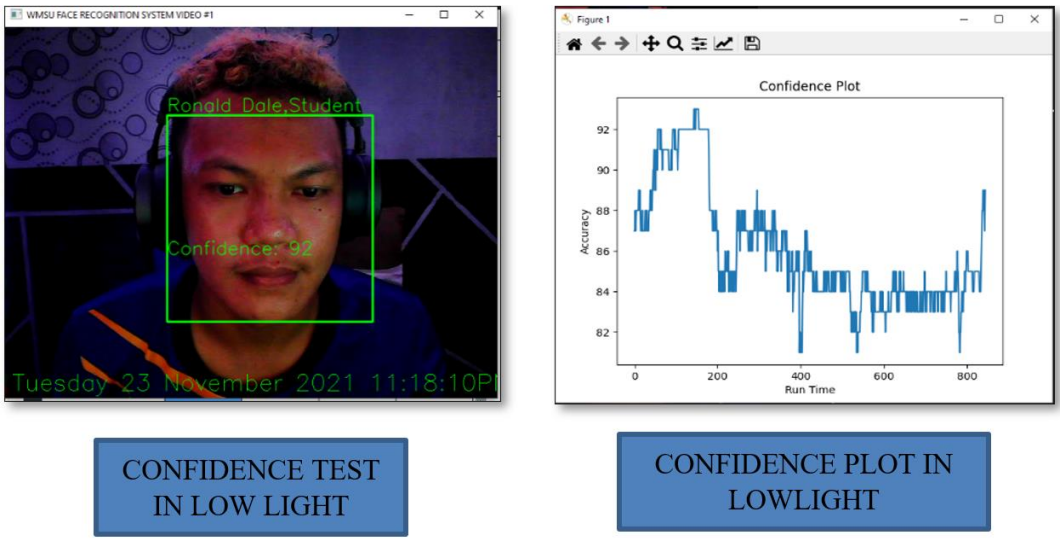


Figure 31: Confidence test in low light

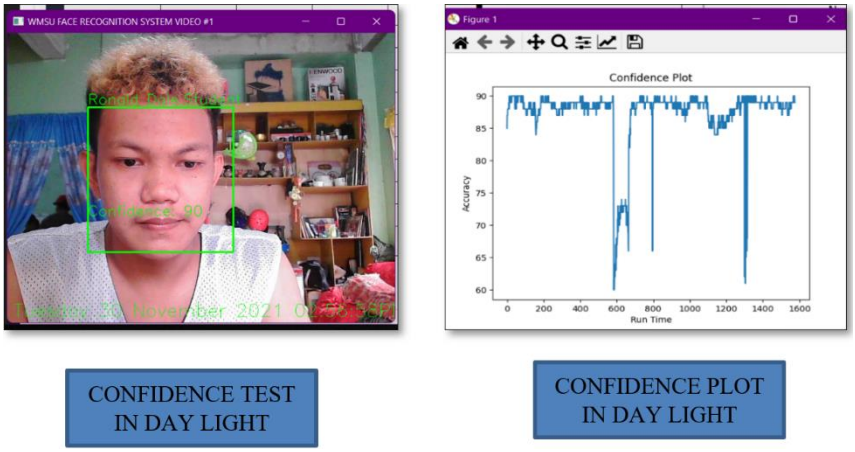


Figure 32: Confidence test in day light

Test	Highest Confidence Rate	Lowest Confidence Rate	Execution time (sec)
Low light	92%	84%	800
Day Light	90%	60%	1600

Table 8: Confidence test for low light and daylight

The figures above compare face recognition and Detection during nighttime using artificial light and natural light daytime. Both tests show no difference at all in terms of recognition confidence rate. The confidence plot in lowlight registered a 92% highest confidence rate and 84% lowest during the system's 800 seconds recognition runtime. The confidence plot in daylight registered a 90% highest confidence rate and 60% lowest during the 1600 seconds recognition runtime.

The confidence test was conducted to show the difference between facial recognition during daytime with natural light and nighttime with artificial light. The test comparison revealed no difference, as both tests resulted in acceptable facial recognition at 90%, making the system functional and reliable in any lighting conditions as long as the camera captures the face.

Testing the System Using Two Cameras Simultaneously

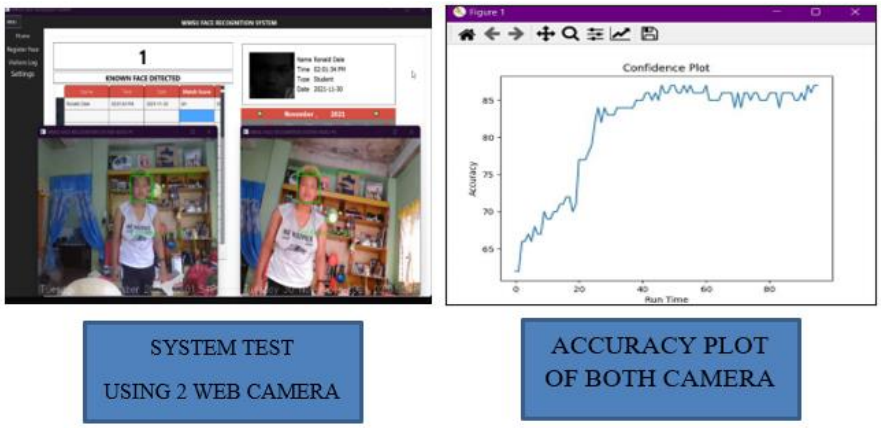


Figure 33: Testing the system with two cameras simultaneously

The system can detect a face with a maximum of 2 meters from the camera. The figures above show the testing of the system using two cameras simultaneously. This testing determines if the system can run using more than one camera. The system successfully recognizes the Subject's face in both cameras during testing. The accuracy plot shows that the system registers a 90% highest confidence rate and below 60% lowest during the 100 seconds recognition runtime.

The test demonstration using two cameras is necessary to determine the capability and stability of the system using two cameras simultaneously. The result shows that the system can use two cameras simultaneously with no difference in its performance and stability as using one camera.

Recognition Accuracy

No. of faces in a live video	Execution Time (sec)	No. of face recognized	Accuracy (%)
1	15	1	100
2	9	2	100
3	11	3	100
4	16	3	75
5	16	4	75

Table 9: Recognition accuracy table

Accuracy Calculation

Accuracy = (No. of Face Recognized / No. Face) * 100

This test proves that having multiple faces in the camera feed would not make the system inoperative. The data shows that even with three faces in the camera feed, the system can still recognize the faces with high accuracy. The system could still detect the face at 4 to 5 faces in the camera feed but with a massive drop in accuracy at 75%.

CHAPTER V
CONCLUSION AND RECOMMENDATIONS

Conclusion

Face recognition is a biometric solution designed to recognize a human face in a way that is less intrusive and requires no contact. Facial recognition technology is more suited to individuals unwilling to collaborate with other biometric identification methods such as fingerprint, iris, or hand scan.

This paper designed and developed a face recognition system using Haar Cascade Classifier and Local Binary Pattern Histogram. The developed system can easily store and enroll subjects' faces and information into the system. The training process time for each new Subject registered only increases by 2-3 seconds, making the system training step very fast. An achievement of up to 90% confidence rate when training 250 images of a person into the classifier. The system can also successfully identify unauthorized individuals visible in the camera that prompts personnel to log their entry. This feature of the system makes it very effective in monitoring people in an organization or in school to provide better security; During the testing, the system was able to perform well in higher light intensities, reaching up to a 92% confidence rate, and dark lighting reaching 90%-92% confidence rate.

Recommendation

The design, implementation, and testing of a face recognition system were completed, and the results of the tests indicated that the system performs satisfactorily. On the other hand, future works can still improve the system. One of which is enhancing the distance of recognition by implementing a more advanced algorithm. Another feature that would be beneficial in future works would be an online service that can manage more than two cameras better to increase the security and effectiveness of the system. Also, implementing the complete web-app-based system in future works would greatly help increase its ease of use.

References

[1] S. Chen, "China to build giant facial recognition database to identify any citizen within seconds," 12 October 2017. [Online]. Available: https://www.scmp.com/news/china/society/article/2115094/china-build-giant-facial-recognition-database-identify-any. [Accessed March 2021].

[2] Koorsen Fire & Security, "THE EVOLUTION OF CCTV SYSTEMS IN COMMERCIAL SECURITY," 05 March 2021. [Online]. Available: https://blog.koorsen.com/the-evolution-of-cctv-systems-in-commercial-security. [Accessed October 2021].

[3] M. M. G. Parekh Payal, "A comprehensive study on face recognition: methods and challenges," 27 March 2020. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/13682199.2020.1738741. [Accessed October 2021].

[4] F. M. Serign Modou Bah, "An improved face recognition algorithm and its application in the attendance management system," 26 December 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590005619300141. [Accessed October 2021].

[5] N. K. Thai Hoang Le, "Applying Artificial Neural Networks for Face Recognition," 2011 03 November. [Online]. Available: https://www.hindawi.com/journals/aans/2011/673016/. [Accessed October 2021].

[6] N. R. Balarini, "Facial Recognition Using Neural Networks over GPGPU," [Online]. Available: http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0717-50002012000300007. [Accessed October 2021].

[7] N. D. A. D. E. P. D. A. Athanasios Voulodimos, "Deep Learning for Computer Vision: A Brief Review," 01 February 2018. [Online]. Available: https://www.hindawi.com/journals/cin/2018/7068349/. [Accessed October 2021].

[8] M. S. Md Nazmus Saadat, "Advancements in Deep Learning Theory and Applications: Perspective in 2020 and beyond," 09 December 2020. [Online].

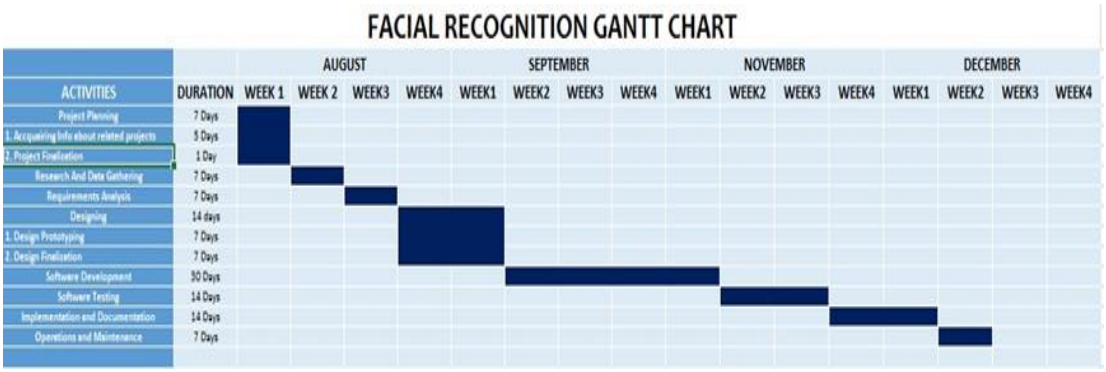
Available: https://www.intechopen.com/books/advances-and-applications-in-deep-learning/advancements-in-deep-learning-theory-and-applications-perspective-in-2020-and-beyond. [Accessed October 2021].

[9] A. H. Ade Nurhopipah, "Motion Detection and Face Recognition for CCTV Surveillance System," 2018. [Online]. Available: https://jurnal.ugm.ac.id/ijccs/article/view/18198. [Accessed October 2021].

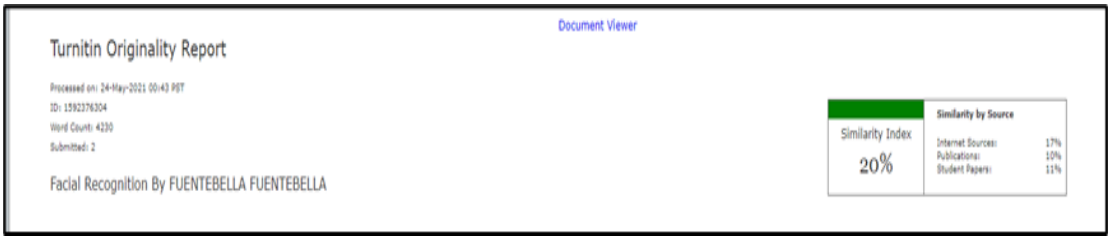
[10] H. M. V. M. A. Mahdi, "Face recognition-based real-time system for surveillance," September 2016. [Online]. Available: Retrieved from www.researchgate.net: https://www.researchgate.net/publication/308387275_Face_recognition-based_real-time_system_for_surveillance. [Accessed October 2021].

[11] L. P. R. S. A. C. C. Alolor, "Student Monitoring System Of Our Lady Of Fatima University Using Face Recognition," 25 December 2014. [Online]. Available: https://www.semanticscholar.org/paper/Student-Monitoring-System-Of-Our-Lady-Of-Fatima-Alolor-Legaspi/9a30e716d28a33c87a8cce15a5e2eb6722d785f1. [Accessed October 2021].

Appendix A Gant Chart



Appendix B Plagiarism Report



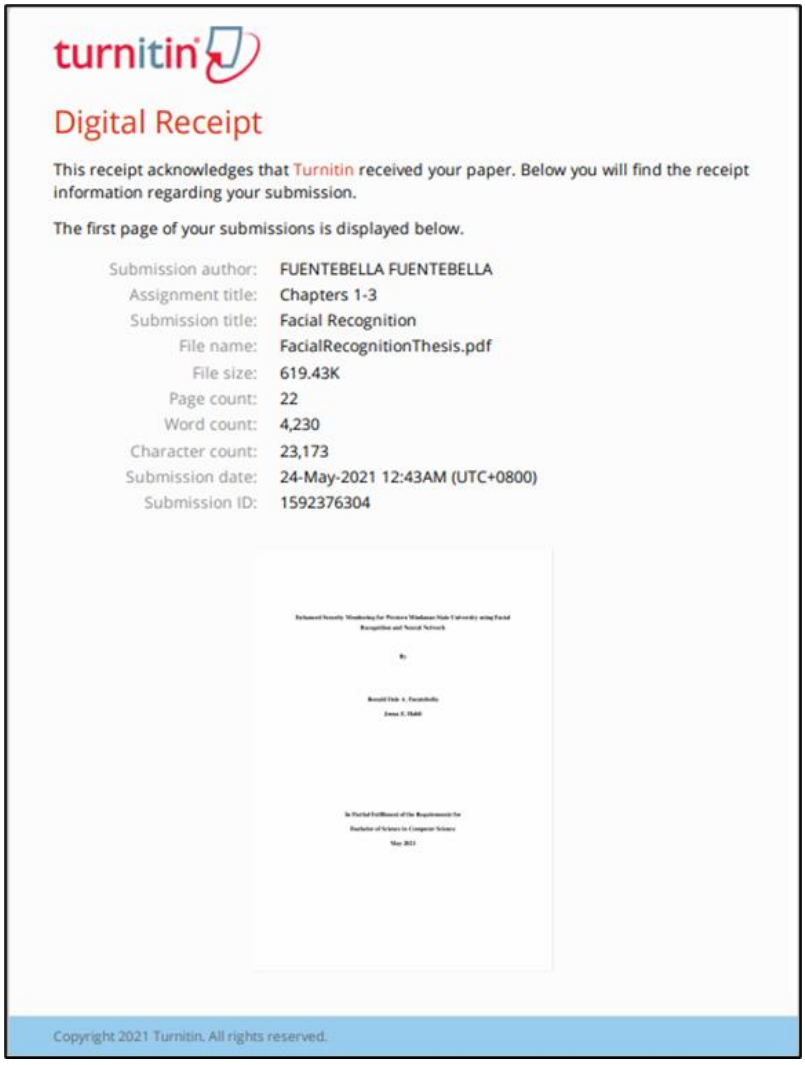
Document Viewer

Turnitin Originality Report

Processed on: 24-May-2021 00:43 PDT
ID: 1592376304
Word Count: 4230
Submitted: 2

Facial Recognition By FUENTEBELLA FUENTEBELLA

Similarity Index	Similarity by Source	
20%	Internet Sources:	17%
	Publications:	10%
	Student Papers:	11%



turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	FUENTEBELLA FUENTEBELLA
Assignment title:	Chapters 1-3
Submission title:	Facial Recognition
File name:	FacialRecognitionThesis.pdf
File size:	619.43K
Page count:	22
Word count:	4,230
Character count:	23,173
Submission date:	24-May-2021 12:43AM (UTC+0800)
Submission ID:	1592376304

Copyright 2021 Turnitin. All rights reserved.

Appendix C Prototype of the System

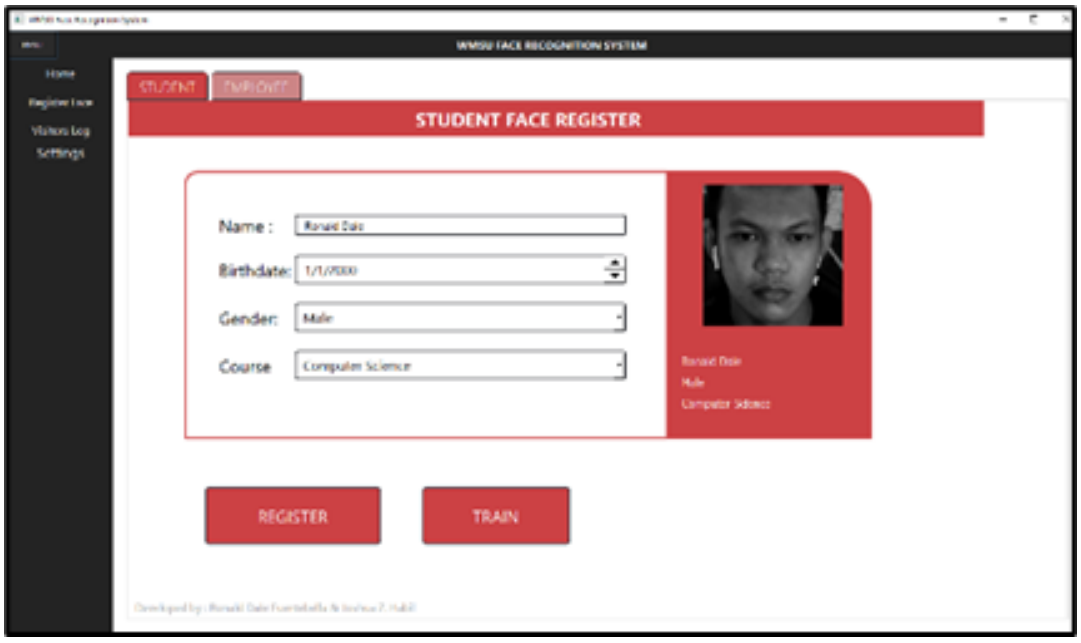


Figure 34: Face Enrolment UI

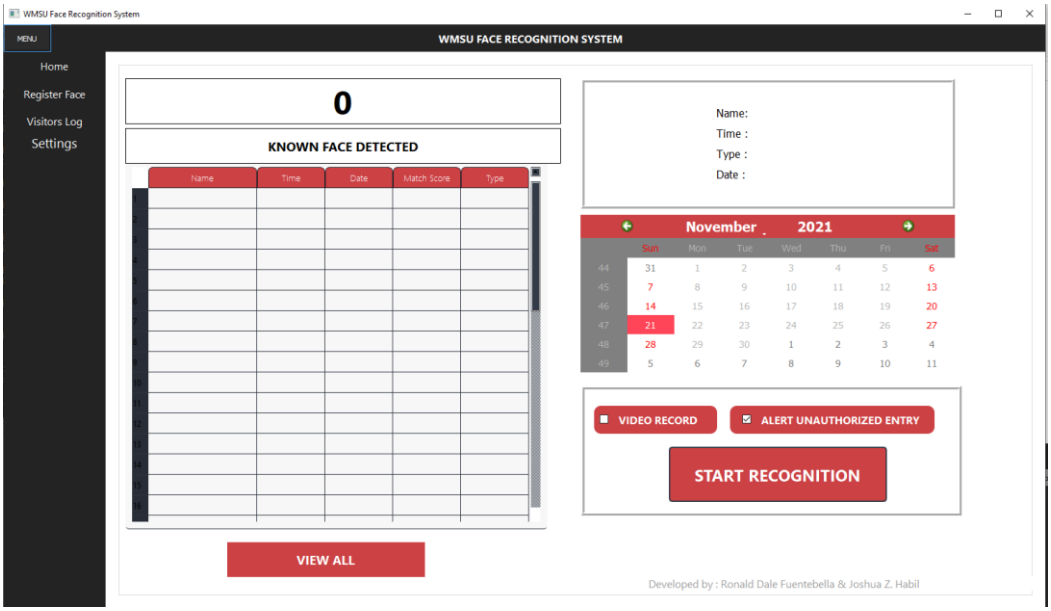


Figure 35: Desktop Application side Home Screen

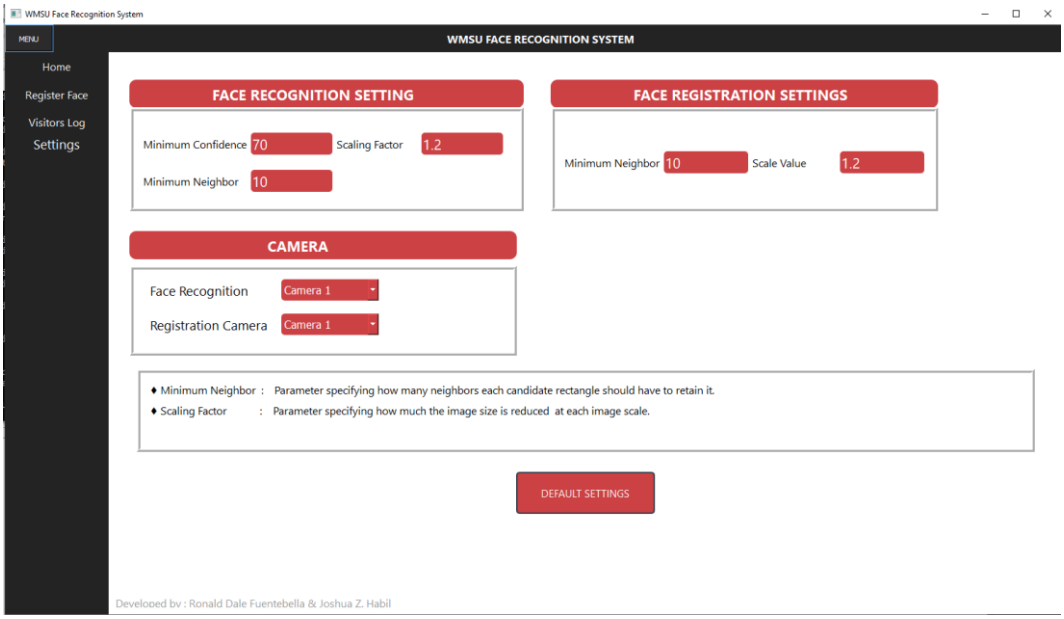


Figure 36: System Settings parameters

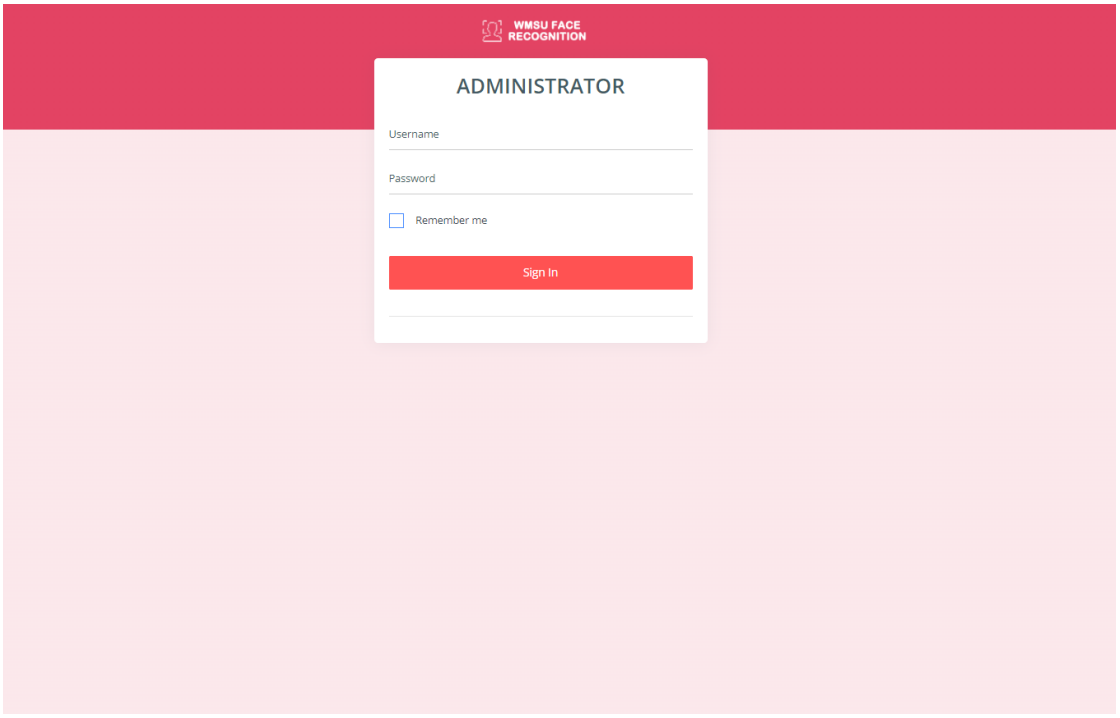


Figure 37: Admin log-in UI

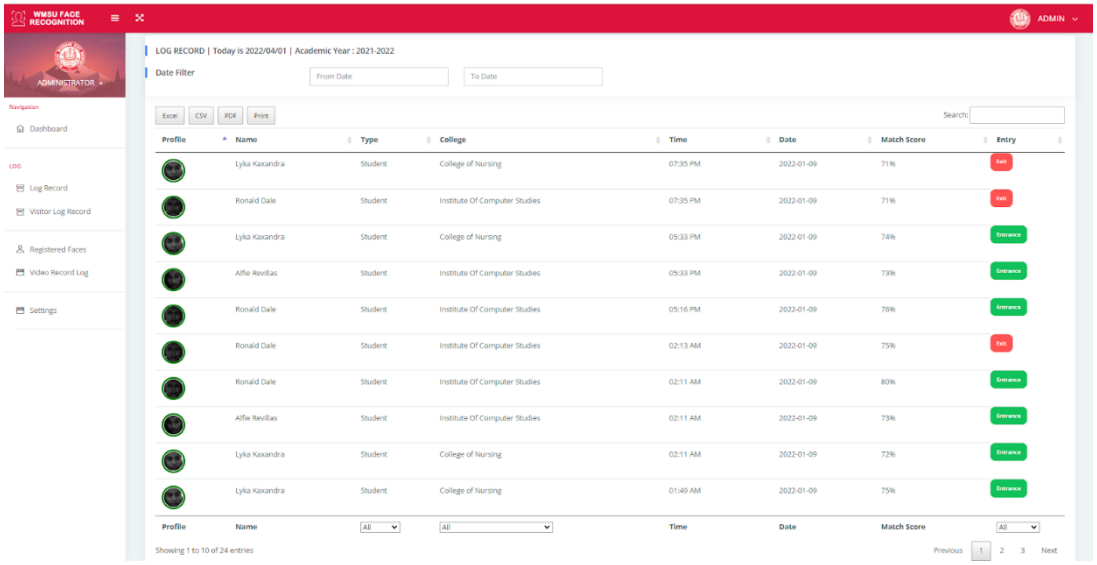


Figure 38: Admin | Dashboard

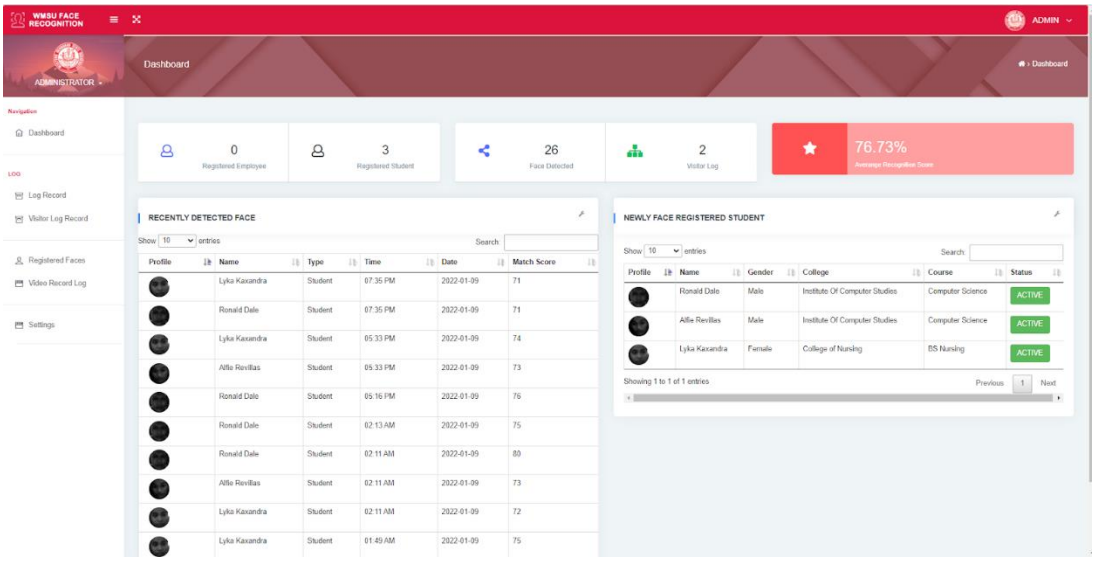


Figure 39: Admin | Known Face Log Record

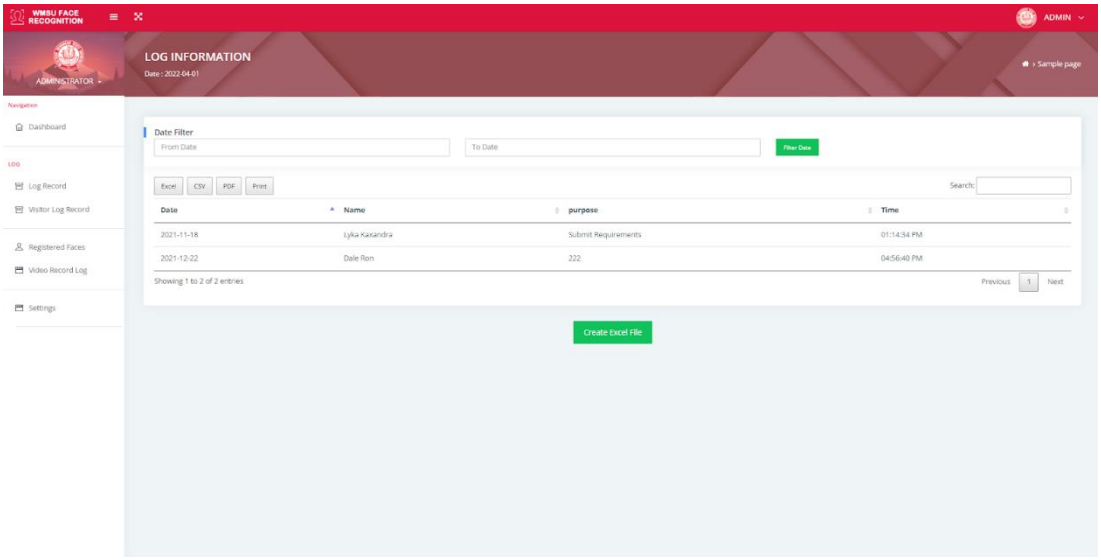


Figure 40: Admin | Visitors Log Record

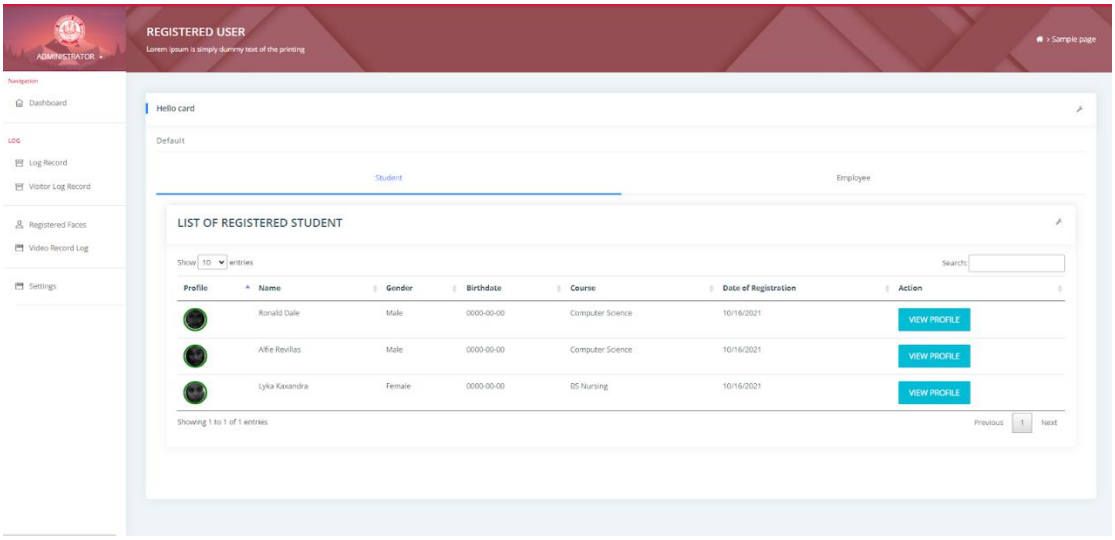


Figure 41: Admin | Registered faces

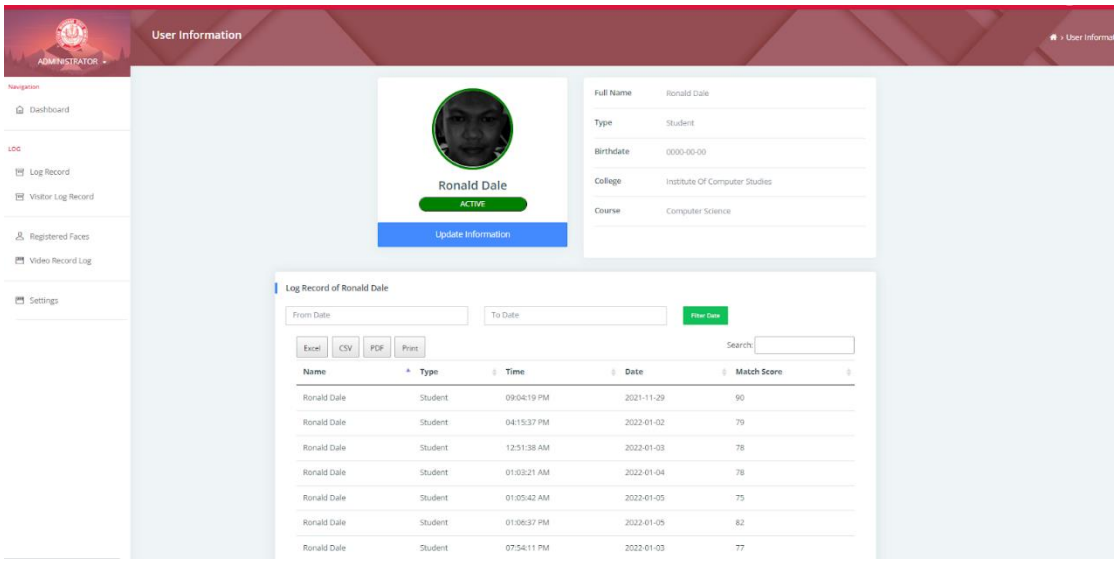


Figure 42: Admin | User Information

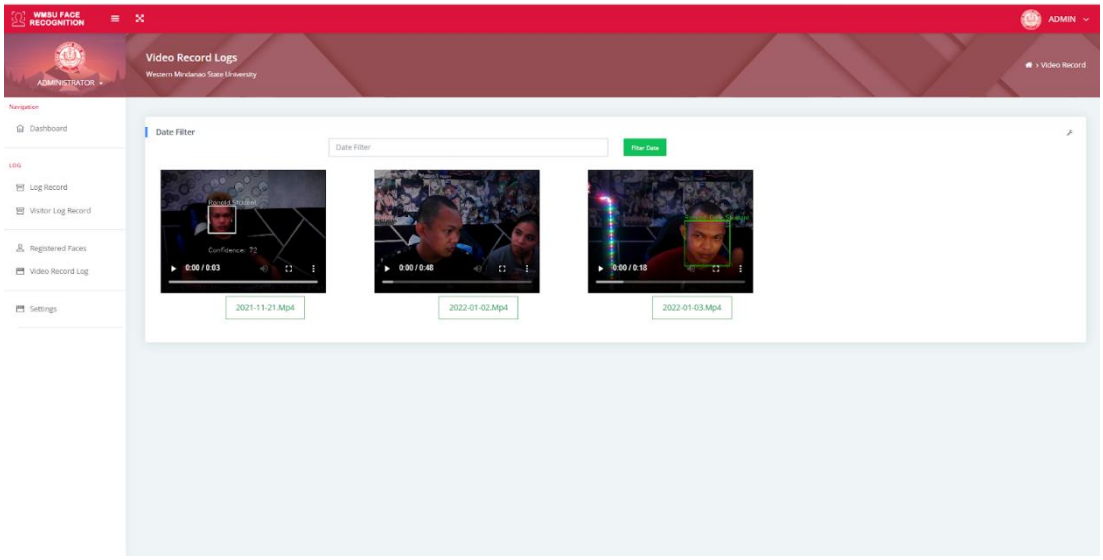
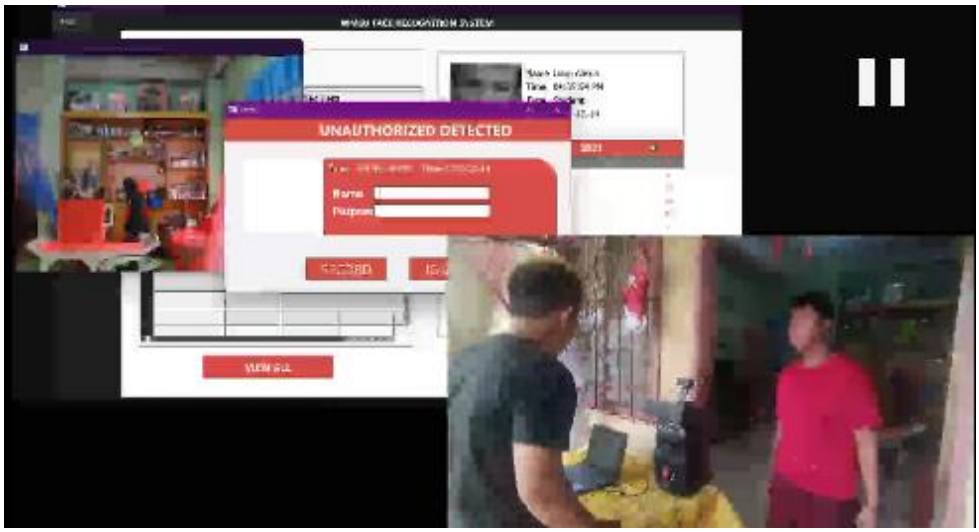


Figure 43: Admin | Video Log Record

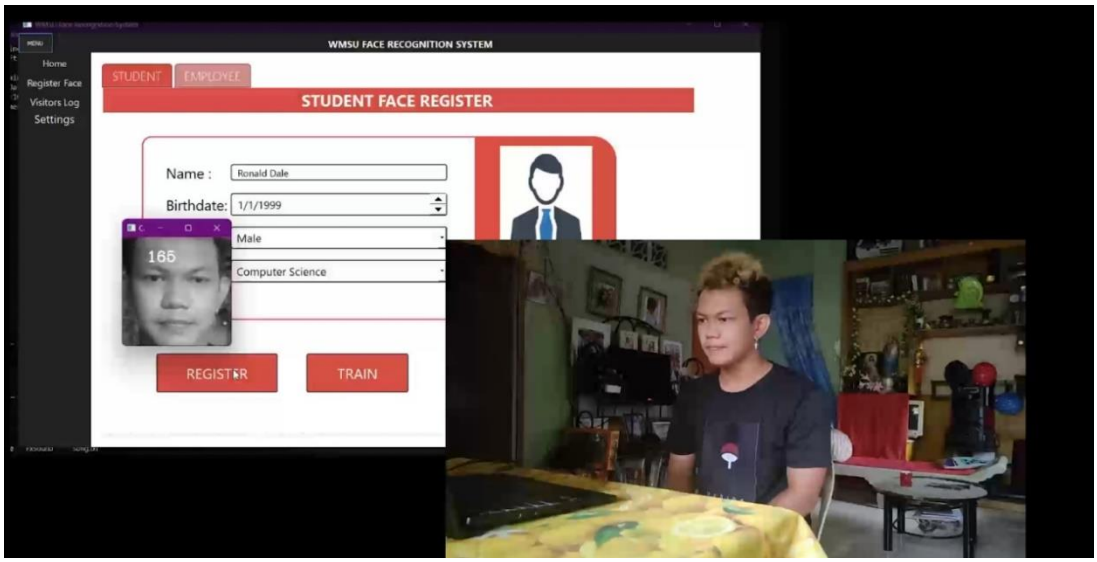
Appendix D Photo Documentation

Test Set

Set up of the system upon entering school gate.



Scenario where an unregistered subject is detected by the system and labelled as unauthorized detected.



The scenario of subjects face registration to the system where the Subject also inputs necessary information

Appendix E User Manual

Facial Recognition System Interface

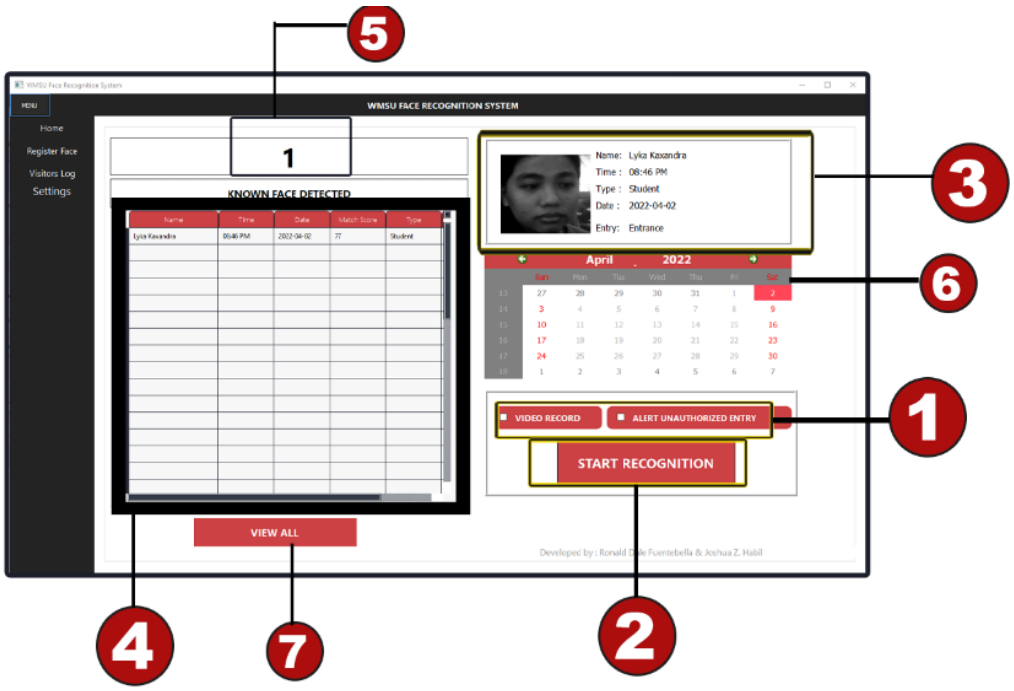
Welcome to the very first facial recognition biometric system of Western Mindanao State University. The Facial recognition System eliminates the hassle of monitoring and logging entries to the school campus bringing ease for students, visitors, employees, and staff of the school. The user manual provides a detailed description of the system's features. This manual will make sure that you'll find this system easy to use.

System Navigation

The main areas of the face recognition system are shown below and are described in detail on the following page.

- Home
- Register Face
- Visitor Log
- Settings

FACE RECOGNITION DESKTOP APPLICATION USER INTERFACE



Home Navigation Page

This page shows the record of all registered faces detected by the system. The date and time of entry are recorded every time a registered face is detected. The system automatically recognizes the face of a registered subject and shows its personal information as well as creating a log record.

[1] Tick Video Record

To record the session.

[1] Tick Alert Unauthorized Entry

To enable alert popup for detected unauthorized entry.

[2] Click Start Recognition Button

To open the selected camera and enable face recognition after gate selection.

[3] Face Detected Information

This section shows basic information like (name, type, entry, recognized time, and date of entry) of the recognized Subject.

[4] Log Record Table

A table that shows all the log records during the day with information such as name, time and date of entry, type, match score, and gate number entry.

[5] Recognition Counter

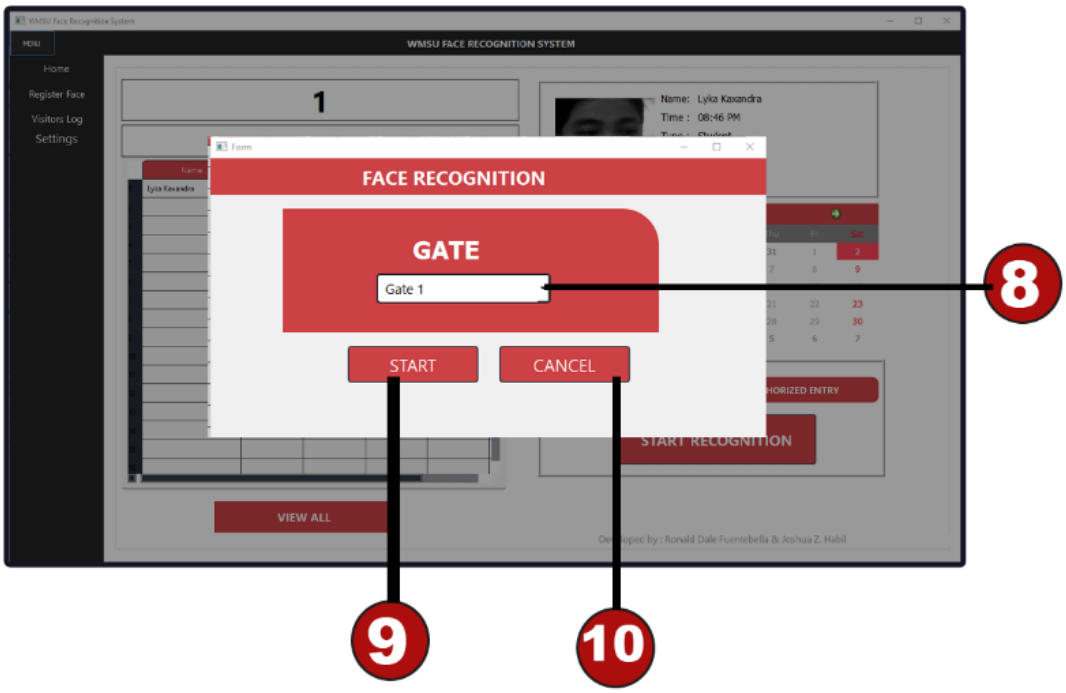
Shows the number of detected subjects during the day.

[6] Calendar

An up-to-date calendar to provide a user ease of checking the date.

[7] Click View All Button

To expand the log record table and view all recognized subjects during the day.

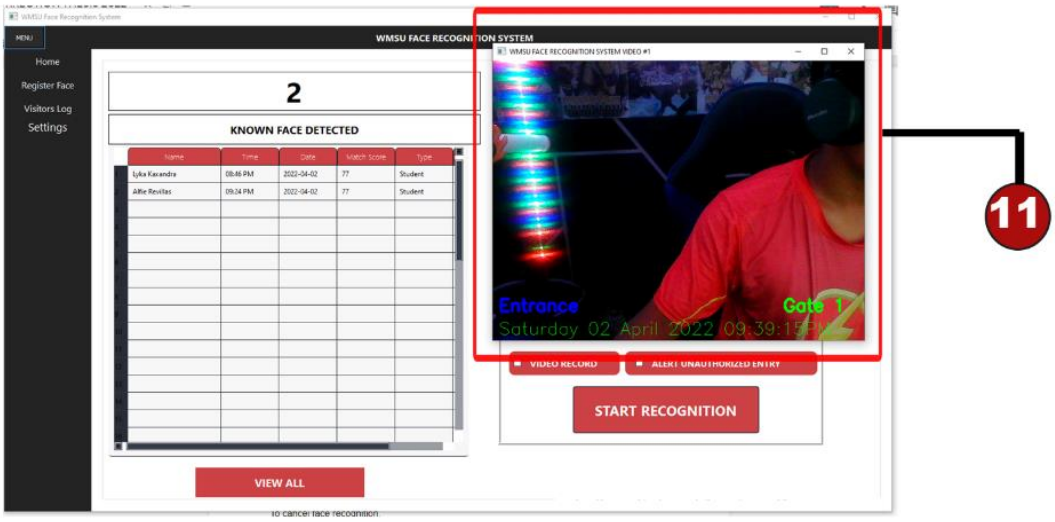


After clicking the start recognition button; a popup will appear prompting user to select which gate is the client located.

[8] Gates - To select which gate to start the face recognition system

[9] Start - To start face recognition in the selected gate.

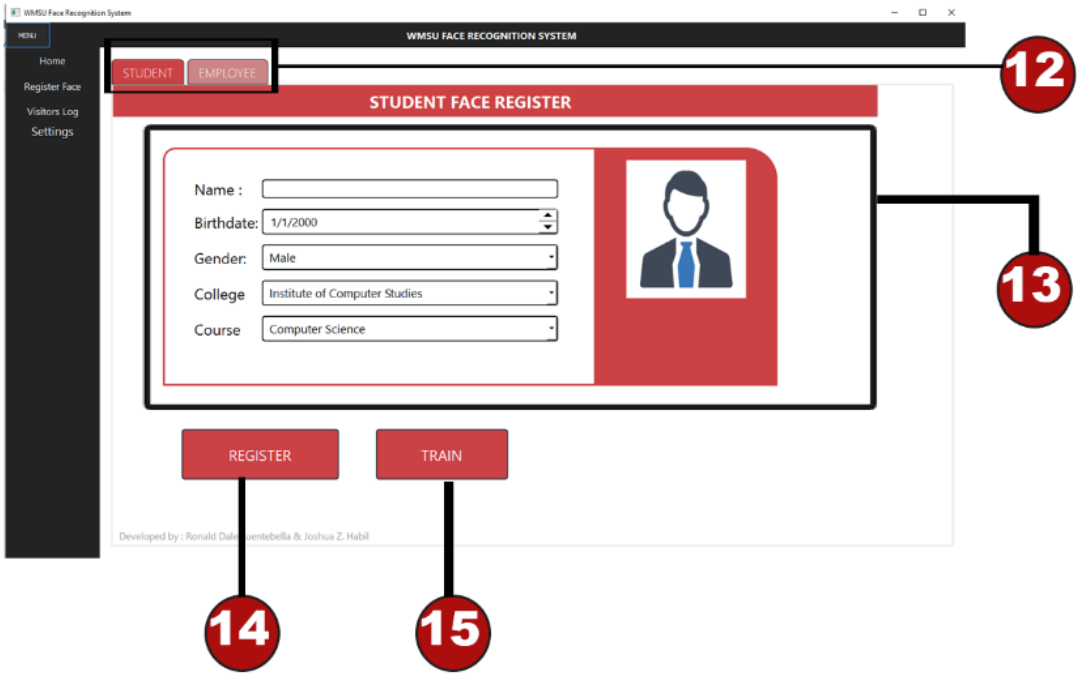
[10] Cancel - To cancel face recognition.



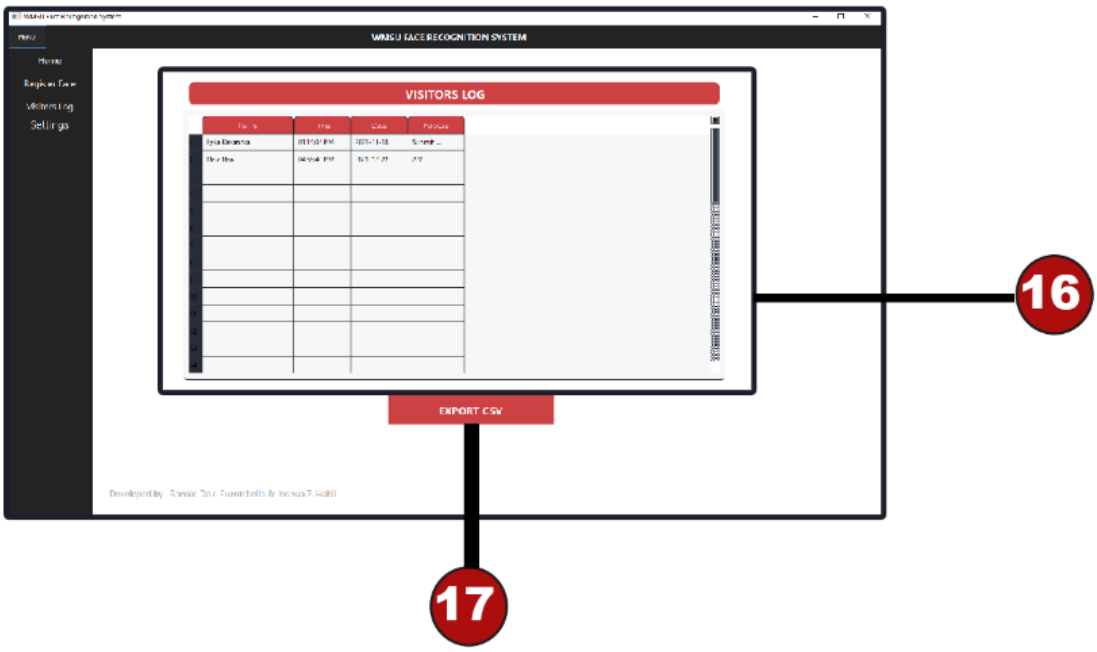
[11] Face Recognition Live Stream Window - Show a live video stream of the subjects providing information such as gate number entry, date of entry, and time of entry.

Register Face Navigation Page

This page is where the registration occurs. There are two types of clients; students and staff or employees. Every type has different information that needs to be imputed upon registration.



[12] Subject selection tab (Student, Employee/staff) - To select the type of Subject.
[13] Required information for registration and training - To input information for subject enrollment in the system.
[14] Click Register Button - To register a subject in the system. All fields in [13] must be filled up. The client (ex. Student, staff, or employee) stands in front of the web camera with a clear view of their face for registration and the registration process will begin. The process will be completed within a few seconds, approximately not more than 5 seconds.
[15] Click Train Button - To train the captured images of the registered Subject and store them in the system.

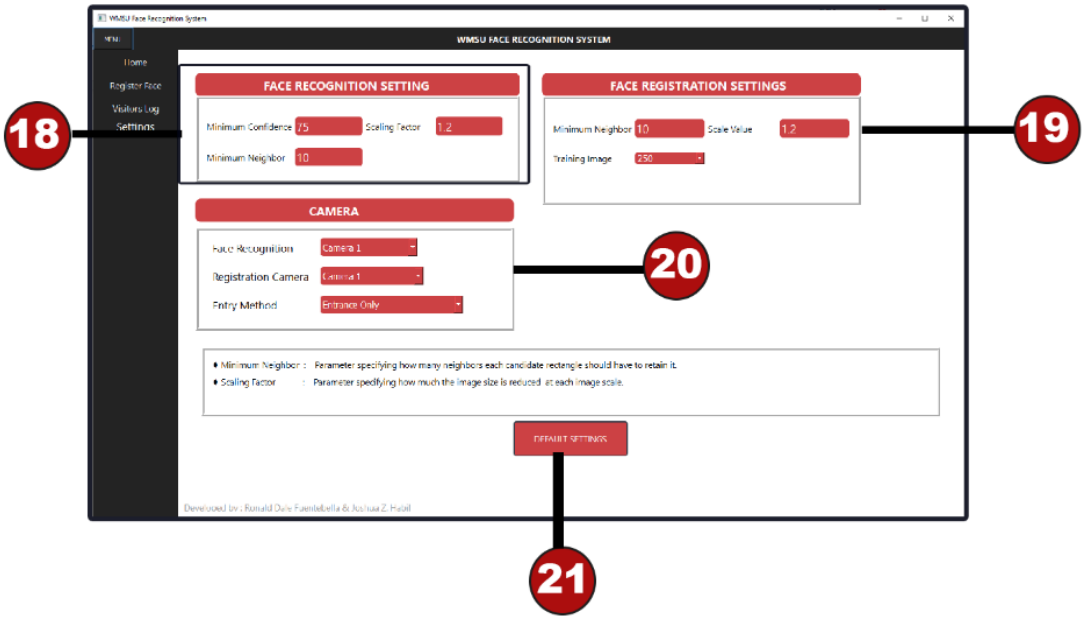


Visitors Log Navigation Page

[16] Visitors Log Table - A table that shows the log record of visitors not registered in the system. The table has important information about the visitor, such as name, time of entry, date of entry, and purpose.

[17] Click Export Button - To download or export the log record into a printable file.

Settings Navigation Page



[18] Face Recognition settings #1

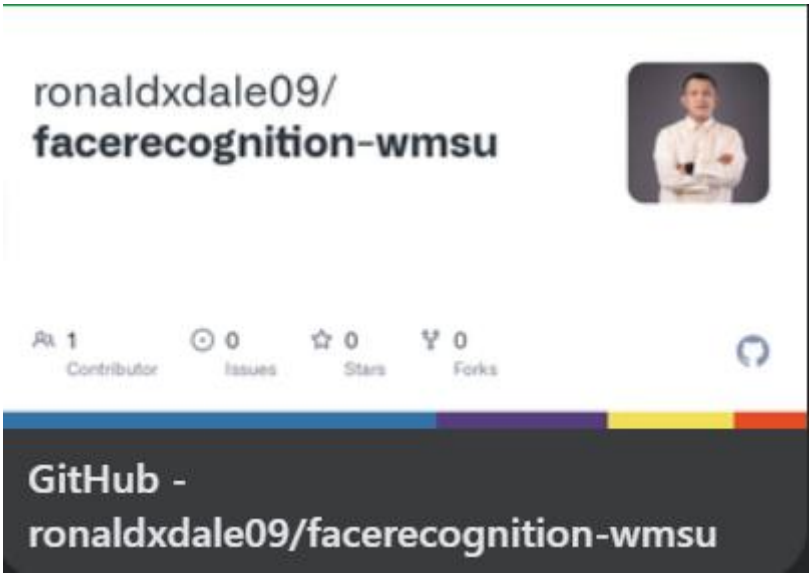
- **Minimum confidence** - the minimum confidence score from 0 to 100 represents the likelihood that the prediction of the subject face is true.
- **Minimum Neighbor** - It is a threshold value that specifies how many neighbors each rectangle should have for it to be marked as a true positive. In other words, let's assume that each iteration marks certain rectangles (i.e. classifies a part of the image as a face). If the subsequent iterations also mark the same areas as a positive, it increases the possibility of that rectangle area being a true positive. If a certain area is identified as a face in one iteration but not in any other iteration, they are marked as false positives. In other words, minNeighbors is the minimum number of times a region has to be determined as a face.
- **Scale Factor -** The smaller the value of scaleFactor, the greater the accuracy and the higher the computation expenses.

[20] Camera settings :

- **Camera Selection -** choose which camera to use (Camera 1, Camera 2, or Both 1 & 2).
- **Registration Camera –** choose what camera to use during face enrollment or subject registration.
- **Entry Method -** Choose whether to enable face recognition at the entrance only or both entrance and exit of the campus gate.

[21] Default Settings - Will revert back to the developer's recommended settings.

Appendix F Source Code



Github - [ronaldxdale09/facerecognition-wmsu (github.com)]

The link above is the source code of both the web app part of the system and the desktop application part.

Desktop Application Part of the system (Face Recognition)

Required Python Package

```
import datetime
from PyQt5 import QtCore
from PyQt5.QtCore import QPropertyAnimation, QThread, Qt, pyqtSignal
from PyQt5.QtGui import QImage, QPixmap
from PyQt5.QtWidgets import QTableWidgetItem
from main import *
import cv2 # OpenCV
import PyQt5.QtCore
from PyQt5.QtWidgets import QMessageBox
import cv2
import os
from PIL import Image #pip install pillow
import numpy as np    # pip install numpy
from datetime import datetime
import mysql.connector
from cv2 import VideoCapture, VideoWriter
from cv2 import VideoWriter_fourcc
from PyQt5.QtGui import QPixmap
import matplotlib.pyplot as plt
```

Function to Generate Student Training Images

To generate the student training images this function require 4 parameters to activate first is the scale factor , minimum neighbor and selected camera, the function also take basic information of the student like (name ,birthdate, course and ID number) to generate 250 training images.

ScaleFactor - A parameter indicating how much the image size is reduced at each image scale. for instance, “1.1” means reduce the size by 10%, increasing the likelihood that the size matches the found feature for detection.

minNeighbors - This parameter affects the quality of detected faces. The higher the value, the fewer the detections, but with higher quality. 3~6 is a good value for face detection.

```
def generateStudent_dataset(self,scale,neigh,selected_camera):

        print('Generating Datasets Started')
        print("Min Neigbor:"+ str(neigh)+" Min Scale: "+
str(scale))
        print('Selected Camera = ' + str(selected_camera))


        name = self.ui.studentName.text()
        birthdate = str(self.ui.studentBirth.date())
        S_birthdate = birthdate.toString("MM/dd/yyyy")


        S_gender = self.ui.studentGender.currentText()
        S_course = self.ui.studentCourse.currentText()
        userType = 'Student'
        status = 'ACTIVE'

        if len(name)==0  or len(S_gender)==0 or len(S_course)==0 :
            self.ui.studentError.setText("Please fill in all
inputs...")
        else:

            connection = mydb.cursor()

            connection.execute("SELECT * from known_faces")
            myresult = connection.fetchall()

            id=1
            for x in myresult :
                id+=1
```

```
            sql= "insert into known_faces
(face_id,name,course,birthdate,sex,userType) values
(%s,%s,%s,%s,%s,%s,%s)"
            val =
(id,name,S_course,S_birthdate,S_gender,userType,status)


            face_classifier =
cv2.CascadeClassifier("haarcascades\haarcascade_frontalface_default
.xml")
            #If face is detected on camera it will capture it then
convert into GRAYSCALE then finally cropped the frontal face
            def face_cropped(img):
                gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
                faces = face_classifier.detectMultiScale(gray, 1.2,
10)
                # scaling factor = 1.5
                # minimum neighbor = 5

                if faces == ():
                    return None
                for (x,y,w,h) in faces:
                    cropped_face = img[y:y+h,x:x+w]
                return cropped_face

            #INSERT DATA IN DATABASE
            connection.execute(sql,val)
            mydb.commit()
            #####
            cap = cv2.VideoCapture(selected_camera)
            img_id = 0
            folder_name= 'face.'+str(id)+"."+userType
            os.makedirs("datasets/"+folder_name)
            while True:
                ret, frame = cap.read()
                if face_cropped(frame) is not None:
                    img_id+=1
                    face = cv2.resize(face_cropped(frame),
(300,300))
                    face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
                    #create folder
                    file_name_path =
"datasets/"+str(folder_name)+"/user."+str(id)+"."+str(img_id)+".jpg
"
                    cv2.imwrite(file_name_path, face)
                    cv2.putText(face, str(img_id), (50,50),
cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,0), 2)

                    cv2.imshow("Cropped face", face)

                #THE MAXIMUM VALUE OF DATASETS BEING GENERATE
                if cv2.waitKey(1)==13 or int(img_id)==250: #13 is
the ASCII character of Enter

            #Display image
                    break
```

```
            cap.release()
            print("DONE DONE DONE")

            ######################## DISPLAY INFORMATION AFTER
REGISTRATION ####################
            profile =
"datasets/"+str(folder_name)+"/user."+str(id)+"."+str(2)+".jpg"
            pixmap = QtGui.QPixmap(profile)
            self.ui.imageLabel.setPixmap(pixmap)
            self.ui.imageLabel.resize(200,200)
            self.ui.imageLabel.setScaledContents(True)

            self.ui.registeredName.setText(name)
            self.ui.registeredGender.setText(S_gender)
            self.ui.registeredCourse.setText(S_course)
            UIFunctions.generateMessage()
```

Function to Generate Employee Training Images

To generate the Employee training images this function also require 4 parameters to activate first is the scale factor , minimum neighbor and selected camera, the function also take basic information of the student like (name ,birthdate, employee Type and ID number) to generate 250 training images.

```
def employee_generate(self,scale,neigh,selected_camera):

            print('Generating Datasets Started')
            print("Min Neigbor:"+ str(neigh)+" Min Scale: "+
str(scale))
            print('Selected Camera = ' + str(selected_camera))


            name = self.ui.employeeName.text()
            E_birthdate = self.ui.employeeBirthdate.date()
            E_birthdate = E_birthdate.toString("MM/dd/yyyy")



            E_gender = self.ui.employeeGender.currentText()
            E_department = self.ui.employeeDepartment.currentText()
            E_type = self.ui.employeeType.currentText()
            E_position = self.ui.employeePosition.currentText()
            userType= "Employee"
            status = "ACTIVE"
```

```
            if len(name)==0  or len(E_gender)==0 or
len(E_position)==0 :
                self.ui.studentError.setText("Please fill in all
inputs...")
            else:

                connection = mydb.cursor()

                connection.execute("SELECT * from known_faces")
                myresult = connection.fetchall()

                id=1
                for x in myresult :
                    id+=1

                sql= "insert into known_faces
(face_id,name,sex,birthdate,college,employeeType,position,userType,
status) values (%s,%s,%s,%s,%s,%s,%s,%s,%s)"
                val =
(id,name,E_gender,E_birthdate,E_department,E_type,E_position,userTy
pe,status)


                face_classifier =
cv2.CascadeClassifier("haarcascades\haarcascade_frontalface_default
.xml")
                #If face is detected on camera it will capture it
then convert into GRAYSCALE then finally cropped the frontal face
                def face_cropped(img):
                    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
                    faces = face_classifier.detectMultiScale(gray,
scale, neigh)


                    if faces == ():
                        return None
                    for (x,y,w,h) in faces:
                        cropped_face = img[y:y+h,x:x+w]
                    return cropped_face

                #INSERT DATA IN DATABASE
                connection.execute(sql,val)
                mydb.commit()
                #####
                cap = cv2.VideoCapture(selected_camera)
                img_id = 0
                folder_name= 'face.'+str(id)+"."+userType
                os.makedirs("datasets/"+folder_name)
                while True:
                    ret, frame = cap.read()
                    if face_cropped(frame) is not None:
                        img_id+=1
                        face = cv2.resize(face_cropped(frame),
(300,300))
                        face = cv2.cvtColor(face,
cv2.COLOR_BGR2GRAY)
                        #create folder
```

```
                        file_name_path =
"datasets/"+str(folder_name)+"/user."+str(id)+"."+str(img_id)+".jpg
"
                        cv2.imwrite(file_name_path, face)
                        cv2.putText(face, str(img_id), (50,50),
cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,0), 2)

                        cv2.imshow("Cropped face", face)

                    #THE MAXIMUM VALUE OF DATASETS BEING GENERATE
                    if cv2.waitKey(1)==13 or int(img_id)==350: #13
is the ASCII character of Enter
                        break


                cap.release()
                print("DONE DONE DONE")
                UIFunctions.generateMessage()
```

Function to Train the Classifier

This function finds all the images in the dataset folder given the name of each image(userType.jpg file), transforms these images into arrays, and then passes them and their ID into recognizer for training. After that, the model creates a format file ".xamp", containing the corresponding histograms and their labels(supervised learning), for further recognition purpose

```
def train_classifier():

        data_dir="datasets"
        path = [os.path.join(data_dir, f) for f in
os.listdir(data_dir)]
        faces = []
        ids = []

        for root, dirs, files in os.walk(data_dir):
            for file in files:
                if file.endswith("png") or file.endswith("jpg"):
                    path = os.path.join(root, file)
                    img = Image.open(path).convert('L')

                    #CONVERTING THE IMAGE INTO NUMPY ARRAY
                    imageNp = np.array(img, 'uint8')
                    #GETTING THE USER ID ON THE IMAGE
```

```
                    id = int(os.path.split(path)[1].split(".")[1])

                faces.append(imageNp)
                ids.append(id)
        print(faces+ids)

        ids = np.array(ids)

        # Train and save
        #LBPH (Local Binary Pattern Histogram) is a Face-
Recognition algorithm it is used to recognize the face of a person
        classifier = cv2.face.LBPHFaceRecognizer_create()
        classifier.train(faces,ids)
        classifier.write("classifier.xml")
        UIFunctions.trainingMessage()
        print("Training Done")
```

Function to Launch Face Recognition

The code snippet below will 7 parameters (`matchscore,scaleVal,neigh,selected_camera,video_rec,notif,selected_entry,gate)` , this will load the 'classifier.xml' and activate the selected camera, if the camera detect a possible face it would draw a green rectangle on the video stream, if the face is recognized by the system the code snippet below would display the name and course of the subject.

```
Def
start_faceRG(self,matchscore,scaleVal,neigh,selected_camera,video_r
ec,notif,selected_entry,gate):
        print("Face Recogniction is starting")
        print("Min Neigbor:"+ str(neigh)+" Min Scale: "+
str(scaleVal)+" Confidence: "+ str(matchscore))
        print('Video Record = ' + str(video_rec))
        print('Selected Camera = ' + str(selected_camera))
        print('Unauthorized Alert = ' + str(notif))
        print('selected_entry = ' + str(selected_entry))


        acc= []
        def draw_boundary(img, classifier, scaleFactor,
minNeighbors, color, text, clf,entrance):
            gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            features = classifier.detectMultiScale(gray_img,
scaleFactor, minNeighbors,
flags=cv2.CASCADE_SCALE_IMAGE,minSize=(64, 64))

            for (x,y,w,h) in features:

                #DRAW A RECTANAGLE TO THE DETECTED FACE
                cv2.rectangle(img, (x,y), (x+w,y+h), color, 2 )
```

```
                id, pred = clf.predict(gray_img[y:y+h,x:x+w])

                #FORMULA TO COMPUTE FOR CONFIDENCE
                confidence = int(100*(1-pred/300))


                now = datetime.now()
                time = now.strftime('%I:%M:%S %p')
                date = now.strftime('%Y-%m-%d')


                # get academic year
                academic = mydb.cursor()
                academic.execute("select * from settings")
                year = academic.fetchall()
                for row in year:
                    acadYear= row[1]


                # get user info in database
                cursor = mydb.cursor()
                cursor.execute("select * from known_faces where
face_id="+str(id))
                person = cursor.fetchall()
                #convert into string

                for row in person:
                    face_id= row[0]
                    name= row[1]
                    course = row[4]
                    userType = row[5]
                    status = row[10]
                    college = row[9]


                name = ''+''.join(name)
                status = ''+''.join(status)

                academic = ''+''.join(acadYear)
                #course = ''+''.join(course)
                # section = ''+''.join(section)

                now = datetime.now()
                #Confidence likelihood of the results being true

                if confidence>matchscore :
                    if(status =='ACTIVE'):
                        cv2.putText(img, name+','+userType, (x,y-
5), cv2.FONT_HERSHEY_SIMPLEX, 0.8, color, 1, cv2.LINE_AA)
                        cv2.putText(img,"Confidence: " +
str(confidence), (x,y+180), cv2.FONT_HERSHEY_SIMPLEX, 0.8, color,
1, cv2.LINE_AA)

self.markAttendance(face_id,userType,name,confidence,college,academ
ic,entrance,gate)
                    else:
                        break
```

```
                else:
                    if notif == 1 :
                        self.unauthorizedDialog(time,date)
                        cv2.putText(img, "UNAUTHORIZED", (x,y-5),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,0,255), 1, cv2.LINE_AA)
            return img




        # loading classifier
        faceCascade =
cv2.CascadeClassifier("haarcascades\haarcascade_frontalface_default
.xml")

        clf = cv2.face.LBPHFaceRecognizer_create()
        clf.read("classifier.xml")

        now = datetime.now()
        date = now.strftime('%Y-%m-%d')
        if video_rec == 1:
            video =
cv2.VideoWriter('recorded/'+date+'.mp4',VideoWriter_fourcc(*'MP42')
,15,(640,480))

          #ENTRY [ENTRANCE AND EXIT]
        Entrance = 'Entrance'
        if selected_entry==1:
            entry = 'Exit'
        else:
            entry = 'Entrance'

        if selected_camera == 2 :
            video_capture1 = cv2.VideoCapture(0)
            video_capture2 = cv2.VideoCapture(1)




            #This loop is for detecting and diplaying the
object/face with rectangle
            while True:
                stream_ok, img1 = video_capture1.read()
                stream_ok, img2 = video_capture2.read()



                img1 = draw_boundary(img1,
faceCascade,scaleVal,neigh, (0,255,0), "Face", clf,Entrance)
                cv2.putText(img1, datetime.now().strftime("%A %d %B
%Y %I:%M:%S%p"),
                            (10, img1.shape[0] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 1)
```

```
                cv2.putText(img1,'Entrance',(10, img1.shape[0] -
50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 3)
                cv2.putText(img1,gate,(500, img1.shape[0] - 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 3)
                cv2.imshow("WMSU FACE RECOGNITION SYSTEM VIDEO #1",
img1)

                #VIDEO FEED #2

                img2 = draw_boundary(img2,
faceCascade,scaleVal,neigh, (0,255,0), "Face", clf,entry)
                cv2.putText(img2, datetime.now().strftime("%A %d %B
%Y %I:%M:%S%p"),
                            (10, img2.shape[0] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 1)

                cv2.putText(img2,gate,(500, img2.shape[0] - 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 3)
                cv2.putText(img2,entry,(10, img2.shape[0] - 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 3)
                cv2.imshow("WMSU FACE RECOGNITION SYSTEM VIDEO #2",
img2)

                if video_rec == 1:
                    video.write(img1)

                if cv2.waitKey(1)==13:
                    break

            video_capture1.release()
            video_capture2.release()

            cv2.destroyAllWindows()
            if video_rec == 1 :
                UIFunctions.videoRecordAlert()

        else:

            video_capture1 = cv2.VideoCapture(selected_camera)
             #This loop is for detecting and diplaying the
object/face with rectangle
            while True:
                stream_ok, img1 = video_capture1.read()

                img1 = draw_boundary(img1,
faceCascade,scaleVal,neigh, (0,255,0), "Face", clf,Entrance)
                cv2.putText(img1, datetime.now().strftime("%A %d %B
%Y %I:%M:%S%p"),
                            (10, img1.shape[0] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 1)

                cv2.putText(img1,'Entrance',(10, img1.shape[0] -
50), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 3)
                cv2.putText(img1,gate,(500, img1.shape[0] - 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 3)
```

```
                cv2.imshow("WMSU FACE RECOGNITION SYSTEM VIDEO #1",
img1)


                if video_rec == 1:
                    video.write(img1)

                if cv2.waitKey(1)==13:
                    break
            video_capture1.release()

            cv2.destroyAllWindows()


        if video_rec == 1 :
            UIFunctions.videoRecordAlert()
```

```
faceCascade = cv2.CascadeClassifier('Cascades/haarcascade_frontalface_default.xml')
```

This is the line that loads the “classifier” (that must be in a directory named “Cascades/”, under your project directory).

Then, we will set our camera and inside the loop, load our input video in grayscale mode (same we saw before).

Now we must call our classifier function, passing it some very important parameters, as scale factor, number of neighbors and minimum size of the detected face.

```
faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.2,
    minNeighbors=5,
    minSize=(20, 20)
    )
```

Where,

- **gray** is the input grayscale image.

- **scaleFactor** is the parameter specifying how much the image size is reduced at each image scale. It is used to create the scale pyramid.

- **minNeighbors** is a parameter specifying how many neighbors each candidate rectangle should have, to retain it. A higher number gives lower false positives.

- **minSize** is the minimum rectangle size to be considered a face.

The function will detect faces on the image. Next, we must "mark" the faces in the image, using, for example, a blue rectangle. This is done with this portion of the code:

```
for (x,y,w,h) in faces:
   cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
   roi_gray = gray[y:y+h, x:x+w]
   roi_color = img[y:y+h, x:x+w]
```

If faces are found, it returns the positions of detected faces as a rectangle with the left up corner (x,y) and having "w" as its Width and "h" as its Height ==> (x,y,w,h). Please see the picture.

Function to Mark Student/Employee attendance

When the system I successfully identify the subject, the code snippet below will get the face_id, userType, name, confidence value, academic year, entrance, and gate entry value

```
def
markAttendance(self,face_id,userType,name,confidence,college,academ
ic,entrance,gate):
        # RECORD AM AND PM
        now = datetime.now()
        time = now.strftime('%I:%M %p')
        date = now.strftime('%Y-%m-%d')

        morningOrAfternoon = ''.join(x for x in time if
x.isalpha())
        if morningOrAfternoon== 'AM':
             morningOrAfternoon ='MORNING'
        elif morningOrAfternoon =='PM':
```

```
            morningOrAfternoon ='AFTERNOON'



        CSV_PATH_ENTRANCE = 'csv_record/'+date+'-'+
morningOrAfternoon+'-Entrance.csv'
        CSV_PATH_EXIT = 'csv_record/'+date+'-'+
morningOrAfternoon+'-Exit.csv'

        if entrance=='Entrance':
            with open(CSV_PATH_ENTRANCE,'r+') as f:
                myDataList = f.readlines()
                nameList = []
                dateList = []
                for line in myDataList:
                    entry = line.split(',')
                    nameList.append(entry[0])
                    #dateList.append(entry[2])
                if name not in nameList :

f.writelines(f'{name},{time},{date},{userType},{confidence},{colleg
e}\n')

self.addAttendance(name,face_id,time,date,confidence,userType,colle
ge,academic,entrance,gate)

self.displayDetected(name,time,date,face_id,userType,entrance)
                    self.loaddata()
        elif entrance=='Exit':
            with open(CSV_PATH_EXIT,'r+') as f:
                myDataList = f.readlines()
                nameList = []
                dateList = []
                for line in myDataList:
                    entry = line.split(',')
                    nameList.append(entry[0])
                    #dateList.append(entry[2])
                if name not in nameList :

f.writelines(f'{name},{time},{date},{userType},{confidence},{colleg
e}\n')

self.addAttendance(name,face_id,time,date,confidence,userType,colle
ge,academic,entrance,gate)

self.displayDetected(name,time,date,face_id,userType,entrance)
                    self.loaddata()
```

Curriculum Vitae

RONALD DALE A. FUENTEBELLA
Baislan
Contact Number:
Email: ronaldxdale@gamil.com

PERSONAL INFORMATION:

Nickname:	Dale
Birthday:	June 17, 1999
Birthplace:	Isabela City Basilan Province
Age:	23 Years Old
Nationality:	Filipino
Religion:	Roman Catholic
Civil Status:	Single
Language Spoken:	Tagalog, Bisaya, Chavacano, English
Father's Name:	Arnold B. Fuentebella
Mother's Name:	Analyn A. Fuentebella

EDUCATIONAL BACKGROUND:

Tertiary:

Western Mindanao State University
Bachelor of Science in Computer Science
Zamboanga City, Philippines
2018 – 2022

Secondary:

AMA Computer College Zamboanga
2017-2018
Furigay College Inc.
2016-2017

JOSUA Z. HABIL
Zone 6 Mercedes Zamboanga City
Contact Number: 09164238083
Email: joshzapantaloonszx@gmail.com

PERSONAL INFORMATION:

Nickname:	Josh
Birthday:	January 17, 1999
Birthplace:	Zamboanga City
Age:	23 Years Old
Nationality:	Filipino
Religion:	Roman Catholic
Civil Status:	Single
Language Spoken:	Tagalog, Bisaya, Chavacano, English
Father's Name:	Jose Wilson B. Habil
Mother's Name:	Magdalena Z. Habil

EDUCATIONAL BACKGROUND:

Tertiary:

Western Mindanao State University
Bachelor of Science in Computer Science
Zamboanga City, Philippines
2018 – 2022

Secondary:

Don Pablo Lorenzo Memorial High school
5th Honorable Mention
2012 – 2015
STI College Zamboanga
With Honors
2015 - 2018