COVID-19 CONTACT TRACING USING ACCESS CONTROL AND FACEMASK RECOGNITION

Azwel Simwinga[1], Jackson Phiri[2]
Computer Science Department
University of Zambia (UNZA)
Lusaka, Zambia
azwelsimwinga@gmail.com

Abstract— Since the outbreak of the Covid-19 pandemic, Facial Recognition technologies have experienced rapid and extensive adoption worldwide. Initially designed for regulating access to specific facilities and ensuring compliance with mask-wearing protocols, these systems now require enhancements to extend their utility beyond the pandemic. This study aims to augment an existing facemask detection system by incorporating future-proof functionalities, particularly Facial Recognition. The inclusion of access control features, such as Facial Recognition, seeks to advance the system's capabilities, allowing for improved user identification and access management. A TensorFlow Lite machine learning facemask detection model was developed, utilizing a dataset collected from GitHub and Kaggle. The dataset consisted of 5,092 photos categorized into three groups: "with_mask," "without_mask," and "mask_worn_incorrect." To ensure accurate model performance, 70% of these images were allocated to the training set, while the remaining 30% were assigned to the test set. Subsequently, a Python application was created to incorporate this robust facemask recognition model. Notably, the Python application goes beyond mere facemask detection by incorporating Facial Recognition capabilities. The Facial Recognition functionality was implemented using the haarcascade_frontalface_default algorithm. Deployed on a Raspberry Pi 4 edge device, the Python application streamlines user registration by assigning each participant a unique ID based on their National Registration Number (N.R.C). The integration of Facial Recognition technology strengthens the system's ability to accurately identify individuals, reducing the chances of impersonation or unauthorized access, while also enforcing facemask regulations. The proposed solution contributes to the ongoing efforts to create safer and more efficient access control systems in a post-pandemic world.

Keywords—Artificial Intelligence, Object Detection, Facial Recognition, Facemask Detection.

I. INTRODUCTION

A pandemic is an epidemic that occurs on a global scale and usually affects a lot of people. Pandemics can be said to occur annually in each of the temperate southern and northern hemispheres, given that seasonal epidemics cross international boundaries and affect many people. However, seasonal epidemics are not considered pandemics.

The Corona virus 2019 (Covid-19) is a contagious disease caused by sever acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first case of Covid-19 was identified in Wuhan City, China, in December 2019 [1]. On January 11th 2020, the first death linked to corona virus was reported when a 61-year-old man passed away [2]. The first case of Covid-19 outside China was officially confirmed in Thailand on 13th January 2020 [3]. By January 21st, 2020, Covid-19 cases were confirmed in South Korea and Japan as well [4]. On the 23rd of January the city of Wuhan and Ezhou were locked down to curb the spread of Covid-19 [5] and it was also confirmed by the World Health Organization (W.H.O) that the virus could be transmitted from human to human. By 26th January 2020, the United States of America (USA) had confirmed a case of Covid-19, and China reported 769 cases of Covid-19. By the 1st of February China reported 14,380 cases with the death toll rising above 300 [6]. The first Covid-19 death outside China was reported in the Philippines on 2nd February 2020 [7]. By 10th February 2020 the death toll of Covid-19 had surpassed that of SARS and Middle East respiratory syndrome with 909 deaths. Africa's first Covid-19 case was recorded in Egypt on 14th February 2020 [8]. On 15th February 2020 the first Covid-19 death outside Asia was confirmed in France when an 80-year-old tourist from Hubei province who had the virus passed on [9]. By 19th February 2020 the death toll due to Covid-19 had surpassed 2000. For the first time since the outbreak began, there are more new cases reported outside China, 459, than in China, 412, on the 26th of February 2020. Brazil also confirmed its first case of Covid-19, marking the first case in South America [10]. Cases of the virus had now been confirmed on every continent except Antarctica. By 28th February W.H.O raised the global risk of spread of Covid-19 from "high" to "very high." [11] and 36117 people in China were reported to have recovered from the disease. W.H.O announced Covid-19 outbreak as a pandemic on 11th March 2020 [12]. On 18th March 2020, Zambia reported first its 2 cases of Covid-19 in Lusaka [13]. The patients were a couple who had travelled to France for holiday. By 19th March 2020 Cases of Covid-19 had surpassed 200,000 globally. It took over three months to reach the first 100,000 confirmed cases and just 12 days to reach the next 100,000. By 29th March 2020 the global death toll had reached 30,000 with 600,000 confirmed cases. As of 9th December 2020, the number of global cases of Covid-19 stands at approximately 68,500,000, with approximately 1,560,000 deaths and approximately 44,200,000 recoveries [14].

Covid-19 can be characterized as a pandemic due to the increase in cases globally. Evidence suggests that Covid-19 is transmitted by people through direct, indirect, or close contact with infected persons or by saliva and respiratory secretions, or through respiratory droplets, which are expelled when an infected person coughs, sneezes, talks or sings. The virus has an incubation period of 5 to 14 days [15]. This incubation period is the time between exposure to the virus and symptom onset. During the incubation period pre-symptomatic patients can be contagious [15]. Therefore, transmission from a pre-symptomatic case can occur before symptom onset. Some people can test positive for Covid-19 from 1 to 3 days before they develop symptoms. The medium time of recovery for mild cases of Covid-19 is approximately 2 weeks, while severe cases of Covid-19 would require 3 to 6 weeks to recover [16]. People with Covid-19 can still infect others even after they stop feeling sick. In response to the first cases of Covid-19 confirmed in Zambia, the Zambian government put in place some measures to curb the spread of the Virus. Some of these measures were: schools, colleges and Universities were closed, travelers entering the country were to be screened, non-essential foreign travel was suspended, public gatherings were limited to 50 people, who were to comply with public health authority guidelines, restaurants were to operate on a take-away and delivery basis, all bars night clubs' cinemas, gyms, and casinos were to be closed, and all international flights were restricted to Kenneth Kaunda International Airport. Citizens were also advised to keep a social distance from one another and maintain high levels of personal hygiene by washing their hands periodically and wearing a face mask to reduce the risk of contracting the virus.

Individuals who are identified as asymptomatic for Covid-19 are typically advised to observe quarantine within their homes. They are directed to medical facilities solely upon the onset of symptoms such as fever, dry cough, breathing difficulties, headaches, and chest pains. The preventive measures of wearing facemasks and practicing social distancing have been widely adopted to mitigate the transmission of Covid-19. Notably, establishments including banks, shopping centers, hospitals, and various public institutions have instituted mandatory guidelines stipulating the use of facemasks by visitors for entry into their premises. Traditionally overseen by human security personnel, the advent of automated Facemask Detection Access Control Systems has facilitated the automation of this responsibility.

With the reduction of Covid-19 Cases and introduction of Vaccines, these systems need to be made future proof in order to keep them relevant. Therefore, extra functionality should be added to them to allow for them to still be useful when the pandemic dies out.

II. LITERATURE REVIEW AND RELATED WORK

The subsequent discourse entails an analysis of pertinent research endeavors within the domain, culminating in a comprehensive examination of related works. This analysis serves as a foundational stepping stone, illuminating the landscape of scholarly contributions that intersect with the subject matter. Through this exploration, we aim to unearth the diverse methodologies, innovations, and findings that have collectively contributed to the ongoing dialogue surrounding the specific focus of interest. By delving into the intricacies of these related works, we seek to establish a contextual framework that enriches our understanding, enhances critical insights, and paves the way for a more informed exploration of the subject's nuances and implications.

The “Spread of Covid-19 Pandemic in Zambia: A Mathematical Model” [17] was an article published to discuss the mathematical model concerning the spread of Covid-19 in Zambia. In order to predict the spread of Covid-19 in Zambia, a multiple regression analysis describing the inter-play of factors influencing the increase in the number of cases is used to formalize the relationship. Exponential graphs regarding the spread of Covid-19 in Zambia were attained to ascertain the accuracy of the model. The mathematical model predicted that 13 people would die within a month if precautionary actions were not taken, and that 65 deaths would occur by the end of the year. The model had limitations such as not being able to trace individuals who have had the infection and recovered if they are immune to getting it once more. The model did not consider post infection immunity.

“Real time Covid-19 facemask detection using deep learning” [18] was a study carried out to assess Machine Learning technologies on how accurately they can detect people wearing masks in pre-recorded videos, photos, or in actual real-time. The project aimed to develop a Graphical User Interface based Automated Facial Recognition as well as Mask Detection System. The algorithms used in the project were Principal Component Analysis (PCA) and HAAR Cascade. A “GREEN” coloured rectangle box would be draw around a face with a mask while a “RED” colour rectangle box would be around a face without a mask. Their model achieved 99% accuracy.

A case study was carried out in Zambia on “The role of forensic pathology in the Covid-19 pandemic in Zambia” [19]. Forensic pathology plays a significant role in safeguarding public health, by identifying and investigating deaths caused by unusual infectious diseases such as Covid-19. Forensic pathology’s participation in mortality and disease surveillance contributes data useful to clinicians and epidemiologists.

The study "Comprehensive Review on Facemask Detection Techniques in the Context of Covid-19" [20] highlights how crucial it is to set up a mechanism to check for facemasks on people's faces in order to ensure public safety and deal with the Covid-19 crisis. The use of facemask detection techniques, a subset of object detection algorithms created for recognizing objects within images, emerges as a critical tactic in this situation. Deep learning has demonstrated substantial effectiveness in facemask detection, which has been credited to its improved feature extraction capabilities over conventional machine learning methods. This has been especially clear among diverse object detection algorithms. However, it is clear that there are many areas in which additional research is needed to enhance the creation of effective facemask detection systems. The study's main goal is to draw researchers' attention by offering a thorough narrative and meta-analytic evaluation of all published papers relevant to facemask detection specifically within the Covid-19 scenario in order to fill in this research gap. The research explores the development of object detection approaches over several decades, taking into account the interaction between facemask detection algorithms and object detection algorithms. This review's extensive study and analysis of the numerous datasets used in different facemask detection experiments is a significant component. The study carefully assesses the performance of several algorithms, using both narrative and meta-analytic techniques to enable a thorough comparison. Ultimately, the paper concludes by not only summarizing the findings but also offering a discourse on the major challenges encountered and identifying potential avenues for future exploration within this specialized domain. Through its thorough investigation and analysis, this study contributes to an enriched understanding of facemask detection technology and its intricate relationship with object detection algorithms, while also addressing its implications in the ongoing battle against the Covid-19 pandemic.

The research paper, "Feasibility and Utility of Facemask Sampling in the Detection of SARS-CoV-2 During an Ongoing Pandemic" [21] addresses the dire need for timely and accessible screening to identify and isolate COVID-19-infected individuals, a crucial element in the global effort to control the virus's spread. The creation of a simple, quick, and affordable COVID-19 testing method is essential to attaining complete screening. In this study, a cohort of 42 patients with COVID-19 positivity and 36 patients with COVID-19 negativity were being examined to determine the feasibility and efficacy of facemask sampling. The procedure involves applying a Steri-StripsTM prototype from 3M to the inner surface of surgical facemasks with loops (Assure). These customized facemasks must be worn by patients for at least 3 hours before being removed and tested for SARS-CoV-2 PCR. Along with this sample strategy, thorough demographic information and specifics regarding the symptomatology of the patients are painstakingly gathered and examined. The study's findings show that the first five days after the onset of symptoms were when facemask sampling showed a higher positive rate. Notably, facemask sampling reliably produces positive SARS-CoV-2 results for patients with nasopharyngeal and/or oropharyngeal swab SARS-CoV-2 PCR Ct values below 25.09. Patients having Ct values of 25.2 or higher, on the other hand, do not show any detectable SARS-CoV-2 presence. The results of the study highlight the effectiveness of facemask sampling as a method for identifying COVID-19, particularly in the early symptomatic stage and in people with high viral loads. The advantage of being able to quickly identify and isolate those who pose the greatest risk of transmission is provided by this capability. Facemask usage is extremely common; therefore, this cutting-edge sampling method has the potential to be used broadly and potentially enable continuous population screening.

In the research work titled "Facemask Detection Algorithm on COVID Community Spread Control using EfficientNet Algorithm" [22], the necessity of automating facemask identification to maintain compliance inside COVID-affected communities is explored as a major worry during the global pandemic. The manual verification technique poses practical difficulties as the enforcement of facemask usage becomes more important. Therefore, in order to shorten and improve the monitoring process, this study responds with an inventive automated approach that makes use of deep learning techniques. This research takes advantage of the growing importance of deep learning algorithms in the field of analysis and detection. The proposed facemask detection technique is carefully optimized to effectively identify facemask compliance by utilizing a Convolutional Neural Network (CNN) driven by the EfficientNet architecture. The base dataset, which consists of 7553 facemask images, is used for both training and validation, and it achieves a commendable accuracy rate of about 97.12%. Importantly, the study's implications go beyond skill level. One important aspect of limiting COVID-19 spread within communities is covered. The proposed technique lessens the workload associated with manual monitoring by acting as a dependable automated solution.

The research, titled "Raspberry Pi Based Crowd and Facemask Detection with Email and Message Alert" [23] examines the effects of the widespread Covid-19 epidemic in 2019 (COVID-19), which has caused considerable destruction in more than 180 countries. As of April 12, 2021, the pandemic's effects included more than 136,772,601 confirmed cases and roughly 2,951,864 fatalities worldwide. Despite the availability of vaccines, the general public has trouble making educated vaccination decisions because of knowledge gaps and anxieties. In order to reduce the spread of the virus, it is still crucial to uphold preventative measures, such as keeping a safe distance and donning facemasks. In accordance with established standards, multiple public health authorities promote maintaining physical separation and wearing facemasks, especially when utilizing public services. This situation has sparked research into novel (computer vision-based) techniques for face mask identification and categorization, with the goal of facilitating the evaluation of societal behavior and aiding in the COVID-19 pandemic mitigation. While research have concentrated on efficient methods to identify and classify face coverings, these frequently presuppose ideal detection circumstances, such as good vision of complex facial characteristics and totally visible faces. Additionally, these techniques mostly target people who are not keeping a safe distance or wearing facemasks. The method that is discussed in this study uses bounding boxes and artificial IDs to track and separate people who have been recognized in a crowd by utilizing the OpenCV object recognition model. This methodology makes it possible to precisely capture recognized people in a changing environment. The results of the OpenCV model are compared to those of more sophisticated models using performance metrics such as Mean Average Precision (MAP), Frames Per Second (FPS), and others while taking structural considerations and boundary restrictions into account. Additionally, the three-dimensional feature space derived from centroid coordinates and bounding box dimensions is used to compute the pairwise vectorized l-2 norm. Real-time alerts are provided via GSM SMS, ensuring prompt notifications, while the Pi camera facilitates data collecting. Python 3 is used to implement the complete framework. In conclusion, the study presents a novel system based on the Raspberry Pi that addresses the issues brought about by the COVID-19 epidemic. By utilizing computer vision technology, this system tackles crowd detection and facemask compliance, providing effective tracking and alert mechanisms. The study's conclusions have ramifications for the way technology-driven solutions to public health emergencies are developing.

The research study "COVID-19 Risk Reduction based YOLOv4-P6-FaceMask Detector and DeepSORT Tracker" [24] discusses the significance of wearing masks as a crucial protective strategy in public situations. Despite the significance of proper facemask usage, there is an absence of study on the identification and tracking of facemasks using image processing methods. This study uses a monocular camera and a deep learning framework to provide a brand-new, highly effective two-stage method for facemask recognition and tracking. The main goal is to utilize video sequences to automatically detect and track facemasks. The development of a novel facemask detection dataset with 18,000 photos is a unique addition of this research. Over 30,000 tightly bound boxes with annotations make up this dataset, which is divided into three different class labels: face masked, improperly masked, and no mask. The YOLOv4-P6-FaceMask detector, which is based on the Scaled-You Only Look Once (Scaled-YOLOv4) object detection model, is trained using the suggested methodology. Additionally, the DeepSORT (Simple Online and Real-time Tracking with a Deep Association Metric) technique is used in the study to track faces. An intriguing innovation surfaces in the use of DeepSORT for tracking faces, employing unique ID assignment to streamline face tracking and establish a repository of no-masked faces. The YOLOv4-P6-FaceMask detector emerges as a highly accurate model, achieving noteworthy metrics including 93% mean average precision, 92% mean average recall, and a real-time processing speed of 35 frames per second (fps) on a single GPU Tesla-T4 graphic card, as evaluated on the proposed dataset. The study compares the results of detection and tracking to other cutting-edge facemask detection and tracking methods in order to demonstrate the effectiveness of the suggested approach. This comprehensive analysis highlights the effectiveness of the YOLOv4-P6-FaceMask detector and the DeepSORT tracker in reducing the hazards associated with COVID-19 by automating facemask monitoring. In conclusion, the research paper offers a significant advancement in the domain of automated facemask detection and tracking. By combining deep learning techniques, novel datasets, and innovative tracking strategies, the proposed framework contributes effectively to addressing the challenges posed by the ongoing pandemic, ultimately fostering safer public environments.

In conclusion, this thorough literature analysis offers insight on the complex web of research surrounding "COVID-19 Contact Tracing Using Access Control and Facemask Recognition." The works that have been assessed as a whole show the complex nature of initiatives made to use technology to stop the epidemic. A sophisticated understanding of the landscape is developed via the examination of various approaches, technical advancements, and theoretical foundations.

III. METHODOLOGY

Data Preparation

GitHub and Kaggle were two of the internet resources used to compile the dataset, which included 5,092 photos. 70% of these images were used in the training set while the remaining 30% were for the testing set. Additional photographs were also taken with a laptop webcam. These pictures were annotated with labels that divided them into three categories: "with_mask," "without_mask," and "mask_worn_incorrect." The items of interest in each image were marked using LabelImg, a widely used program for image annotation, as part of the labeling process. The Pascal VOC format, a widely used standard for labeling pictures in object identification applications, was then used to save the annotations.

Each annotated picture in the Pascal VOC format is specified by its filename, size, and a list of annotated objects according to a particular XML structure. The XML file contains details on each object, including its class label, bounding box coordinates, and extra properties as needed. The annotated data may now be represented consistently and systematically, which makes it simpler to read and use during training.

By utilizing the Pascal VOC format, the annotation files provide vital information to train the machine learning model accurately. They serve as ground truth references, enabling the model to learn and predict the desired outcomes effectively. The structured nature of the format ensures that the model can process and interpret the annotations consistently, facilitating efficient training and evaluation of the model's performance.

In summary, the Pascal VOC format plays a crucial role in the annotation process by providing a standardized representation of the labeled data. It allows for clear and concise communication of the object annotations, supporting the development of accurate and robust machine learning models for object detection tasks.

Face mask detection Model Training

A robust cross-platform deep learning framework called TensorFlow Lite (TFLite) was created to make it possible to deploy pre-trained TensorFlow models on low-performance edge devices. In comparison to conventional computers, these edge devices, such the Raspberry Pi 4, have less computing power. By transforming TensorFlow models into a specific format that may be tailored for performance or storage, TFLite overcomes this barrier and enables effective execution on edge devices.

Model accuracy and execution performance must be traded off when using a TFLite model architecture. Models with more precision run more slowly, while those designed for speed may give up some accuracy. Finding the ideal model architecture for a given application requires striking the correct balance between accuracy and speed. A TensorFlow Lite (TFLite) model can be developed through two distinct routes. One avenue involves training the model directly within the TensorFlow Lite framework. Alternatively, an existing Keras model can be constructed and subsequently converted to the TFLite format, enabling compatibility and optimization for deployment on resource-constrained platforms.

According to the study "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" [25] by Mingxing Tan and Quoc V. Le, EfficientNetB2 is a Keras model architecture that is a member of the EfficientNet family. The EfficientNet model family is well known for its effectiveness and superior performance across a wide range of computer vision tasks.

There is a trade-off between model accuracy and execution performance when converting a Keras model like EfficientNetB2 to a TFLite model. The mid-sized EfficientNet variation, EfficientNetB2, strikes a balance between precision and model size. Compared to EfficientNetB0 and B1, it is larger and more precise. However, it uses fewer resources than EfficientNetB7, B6, B5, B3, and B4. This makes EfficientNetB2 a good candidate for Raspberry Pi 4 operation, particularly when converted to TFLite.

Overall, when converted to TFLite, EfficientNetB2 offers a solid balance between execution performance and accuracy, making it a great option for deploying facial mask recognition software on limited-resource gadgets like the Raspberry Pi 4. As we can see in figure 1, with EfficientNetB2, we were able to maintain the model's efficiency for real-time inference on the Raspberry Pi 4 while achieving a respectable degree of accuracy. The table provides a concise overview of different variants of the EfficientNet model along with their corresponding characteristics. Each variant is listed along with its model size in terms of parameters, top-1 accuracy achieved on the ImageNet dataset, top-5 accuracy on ImageNet, and a general indication of execution speed. The variants range from B0 to B7, with increasing model sizes and improving accuracy. Execution speed varies across the models, with some being faster (Fast) and others slower (Moderate, Slow) based on the task at hand. This table allows for quick comparison and selection of the appropriate EfficientNet variant for specific use cases.

Variant	Model Size	Top-1 Accuracy	Top-5 Accuracy	Speed
EfficientNetB0	5.3 m	77.1%	93.5%	Fast
EfficientNetB1	7.8 m	79.0%	94.8%	Fast
EfficientNetB2	9.2 m	80.2%	95.5%	Fast
EfficientNetB3	12.2 m	81.6%	96.1%	Fast
EfficientNetB4	19.4 m	82.9%	96.7%	Moderate
EfficientNetB5	30.4 m	83.7%	97.1%	Moderate
EfficientNetB6	43.0 m	84.4%	97.4%	Slow
EfficientNetB7	66.3 m	84.8%	97.6%	Slow

Figure 1: Comparison of EfficientNet Variants: Model Size, Accuracy, and Execution Speed

The DataLoader class, which handled the processing of the images and their associated annotations, was used to load the training data. There were three classes in the dataset: "without_mask," "mask_worn_incorrect," and "with_mask." For face mask identification tasks, these class labels must represent the various states of face masks. The initialization and training phases of the EfficientNetB2 model were completed.

After training each model variant, its accuracy was evaluated using the DataLoader's evaluate() method. The evaluation was performed on a separate validation set, which was not used during training. The accuracy metric provided insights into the model's performance and generalization ability.

The tf.saved_model.save() function was used to save the trained EfficientNetB2 model as a Keras model. The TFLiteConverter was then used to change the Keras model into a TFLite model. The converter was configured to use the quantization default optimizations, which are essential for shrinking the model's size and making it appropriate for deployment on devices with limited resources.

Facial Recognition

The Haar Cascade method, especially the haarcascade_frontalface_default algorithm, was used for the system's Facial Recognition training. An Object Detection Algorithm that is frequently used for locating faces in still photos or live movies is Haar Cascade. Paul Viola and Michael Jones first described the approach in their 2001 research work , "Rapid Object Detection using a Boosted Cascade of Simple Features."[26].

To find features, the Haar Cascade algorithm looks for edges or lines. By adding the pixel intensities inside consecutive windows, it determines Haar characteristics. Instead of processing each pixel separately, integral pictures are used in calculation to improve efficiency. The total of the pixel values in various rectangular areas of the image is represented by these integral images. The detecting procedure is accelerated greatly by this technology.

During training, the algorithm employs the AdaBoost algorithm, which selects the best features from the integral images and trains them. It iteratively adjusts the weights of different features to focus on more informative ones. This boosting process improves the algorithm's accuracy in detecting faces.

The haarcascade_frontalface_default is a specific Haar Cascade classifier designed for detecting frontal faces in images. It has been pre-trained on a large dataset of faces and is widely used for face detection tasks.

In the system implementation, the TensorFlow Lite (TFLite) file, which contains the trained machine learning TFLite model converted from EfficientNetB2 architecture, is utilized. A TFLite file is a specialized format that allows for optimized deployment of models on low-performance edge devices like the Raspberry Pi. In the Python application running on the Raspberry Pi, the TFLite file is loaded and used for inference tasks. The TFLite file contains all the necessary information and weights from the trained model, enabling the Raspberry Pi to perform face mask detection tasks efficiently.

By combining the Haar Cascade algorithm for face detection and the TensorFlow Lite file for efficient model execution, the system can accurately detect and recognize faces and facemasks in real-time on the Raspberry Pi.

IV. RESULTS AND DISCUSSION

The EfficientNetB2 model was developed and tested in a variety of settings. Batch size, learning rate, and epoch count are among the crucial variables. Before and after the model was converted to TFLite, the correctness of the model was noted. The experiment was run on a dataset that would be used to identify facial masks.

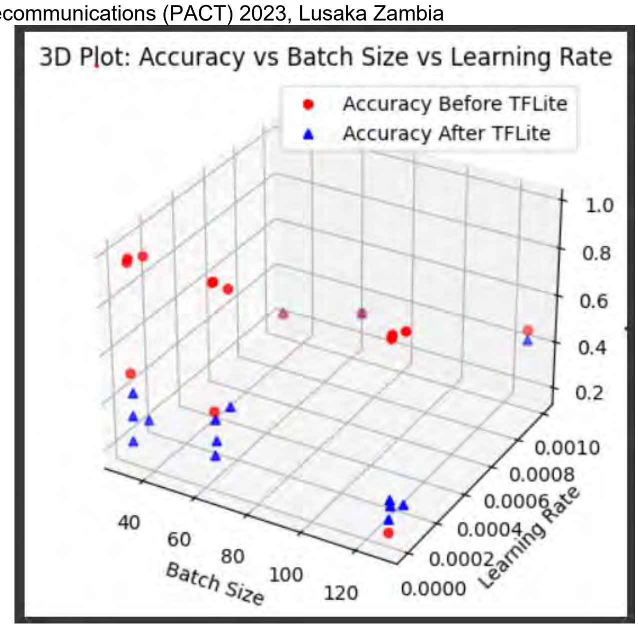

Figure 2: 3D Analysis of Accuracy in relation to Batch Size and Learning Rate for EfficientNetB2 before and after TFLite Conversion.

The graph presented in figure 2 illustrates a three-dimensional analysis showcasing the interplay between accuracy, batch size, and learning rate for the EfficientNetB2 model. The data points are represented in red spheres, denoting accuracy before TFLite conversion, and blue triangular markers, representing accuracy after TFLite conversion. The x-axis corresponds to batch size, the y-axis represents the learning rate, and the z-axis signifies the accuracy value.

The graph's primary focus is to discern patterns and relationships among these factors. It becomes apparent that batch size and learning rate are pivotal in influencing accuracy trends. Notably, the comparison of accuracy before and after TFLite conversion offers insights into the impact of model optimization on overall performance.

This visualization provides a comprehensive understanding of how adjustments in batch size and learning rate can potentially affect accuracy levels, while also highlighting the significance of TFLite conversion in enhancing model efficiency.

Model	Keras	TFLite	Batch Size	Learning Rate	Epochs
EfficientNetB2	0.909	0.424	64	0.0001	20
EfficientNetB2	0.943	0.258	32	0.0001	30
EfficientNetB2	0.944	0.239	128	0.0001	15
EfficientNetB2	0.984	0.332	64	1e-05	20
EfficientNetB2	0.979	0.424	32	1e-05	30
EfficientNetB2	0.979	0.289	128	1e-05	15
EfficientNetB2	0.338	0.338	64	0.001	20
EfficientNetB2	0.23	0.239	32	0.001	30
EfficientNetB2	0.467	0.424	128	0.001	15
EfficientNetB2	0.983	0.424	64	5e-06	20
EfficientNetB2	0.966	0.33	32	5e-06	30
EfficientNetB2	0.966	0.317	128	5e-06	15
EfficientNetB2	0.461	0.276	64	1e-07	20
EfficientNetB2	0.513	0.225	32	1e-07	30
EfficientNetB2	0.183	0.239	128	1e-07	15

Figure 3: Table of Training and Conversion of EfficientNetB2 Model

The experiments involving the EfficientNetB2 model aimed to identify the optimal parameter combination that maximizes accuracy while considering TFLite conversion effects. The table in figure 3 summarizes accuracy measurements before and after conversion, along with batch size, learning rate, and epochs.

Results consistently show reduced accuracy after TFLite conversion compared to before. The choice of batch size significantly affects post-conversion accuracy, with smaller batches mitigating accuracy loss. Learning rate impact is less conclusive, and epoch variations do not consistently impact accuracy.

A potential favorable parameter set for higher accuracy after TFLite conversion involves the EfficientNetB2 model, batch size of 64, learning rate of 0.001, and 20 epochs. Selecting optimal parameters remains context-dependent, considering dataset, model complexity, resources, and the accuracy-speed trade-off.

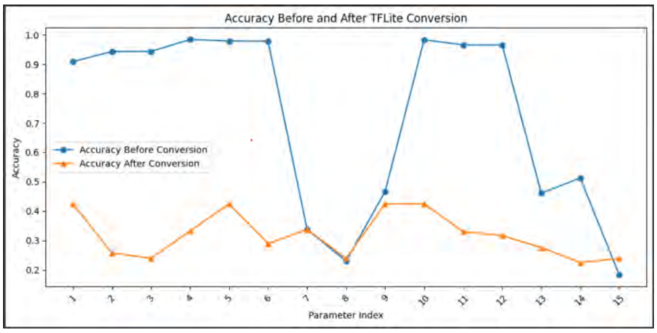

Figure 4: Graph: Comparative Analysis of Accuracy Before and After TFLite Conversion

The graph in figure 4 presents a comparative analysis of accuracy metrics before and after the conversion to TFLite across various parameter settings. The x-axis represents different parameter configurations, while the y-axis denotes the accuracy values achieved. The orange markers indicate accuracy before TFLite conversion, and the blue markers signify accuracy after conversion.

Using a GUI-based Python application running on a Raspberry Pi 4, the Facial Recognition attendance and Facemask Recognition systems were integrated. A 720p USB Webcam serves as the application's video input source. Users may sign up by entering their username, identity number (such as their National Registration Number, Man Number, or Student ID Number), and five photographs of their faces for facial recognition training.

The application implements a robust and reliable process to validate and grant access to registered users. The Raspberry Pi 4's USB Webcam is initially used by the Facial Recognition function to record a live video feed of the user's face. The Facial Recognition algorithm then analyzes this video stream to identify whether the individual is already registered with the system. If the individual is registered within the system, the Facemask Recognition function becomes active, and the Facial Recognition program is shut off if the user is identified. The Facemask Recognition only shuts off when the user is validated to be having their mask on correctly.

On the other side, if the Facial Recognition function is unable to identify the user, the program will continue to run and designate the person as "Unknown." Access is only granted to a user if they are first recognized and are wearing a mask.

Figures 5 and 6 illustrate the utilization scenarios for both system users and system administrators. With elevated privileges, system administrators as seen in figure 5, possess the capability to include new users within the system. On the other hand, users' as seen in figure 6, primary interaction with the system revolves around initiating Facial Recognition and Facemask Detection procedures to attain access clearance.

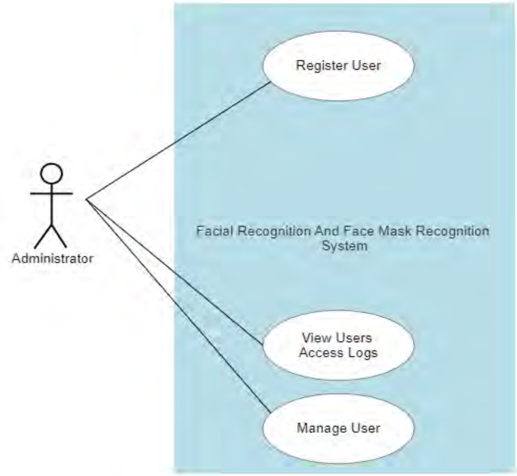

Figure 5: Administrator Use Case Diagram

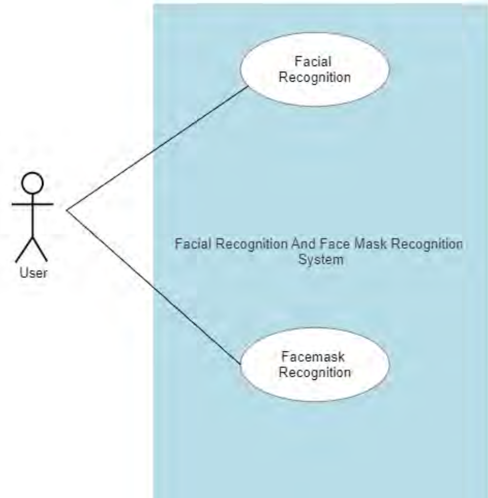

Figure 6: User Use Case Diagram

The program logs each user's access times and stores them in a database. For security reasons, this log also includes the user's name, access date and time, and a photo of the person. To store and handle these logs, the Python program interacts with a PHP Laravel web site. An Application Programming Interface (API) enables this communication. Users' information is communicated to the web application via the API when they register on the Python application. User logs are tracked and managed using a centralized interface provided by the Laravel web application. Figure 7 illustrates the graphical user interface of the Python application

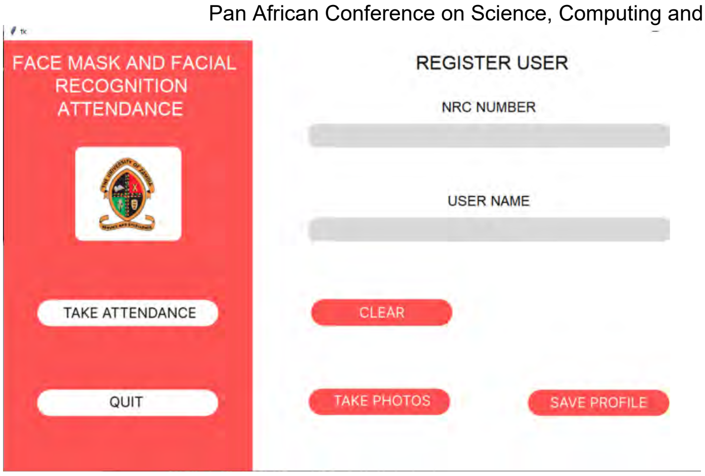

Figure 7: Python App Graphical User Interface

In conclusion, user registration, access verification, and logging are managed by the Python program running on the Raspberry Pi 4 in conjunction with the Laravel web application. The application's workflow makes it easier to integrate Facemask with Facial Recognition. The system's total performance is governed by the accuracy of the Facemask Recognition model, which is affected by variables including the use of TFLite, the architecture used, and the caliber of the dataset.

V. CONCLUSION

In conclusion, the software part of the project is nearly complete, with only a few additional quality-of-life features remaining. One of these features includes developing a web application that allows users to view the attendance log. The current system design is depicted in Figure 8. However, it should be noted that this design is subject to further modifications, as the ideal design aims to conduct Facial Recognition and registration directly on the web application, while sending user details to the Raspberry Pi 4. This design would be particularly advantageous in scenarios where multiple Raspberry Pi 4 devices are being utilized.

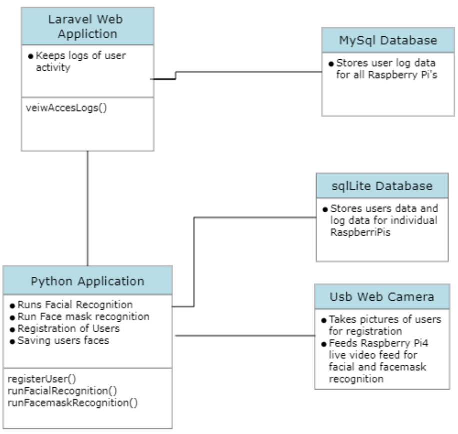

Figure 8: System Context Diagram

Following that, the primary focus would shift towards the hardware aspect, which involves procuring the necessary materials for constructing the access control doors. While the Raspberry Pi 4 has demonstrated effectiveness in running the machine learning application, opting for a Google Coral microcomputer would provide even greater power and efficiency. Additionally, the hardware section of the project includes acquiring a long-range temperature sensor and the sensor gun. These components will further enhance the functionality of the system. Figure 9 illustrates the current design of the proposed door for access control. The USB web camera will be mounted at the top and will be used to feed the Raspberry Pi 4 with input. The door is to be mechanically control by a motor depending on the feedback it receives from the Raspberry Pi 4.

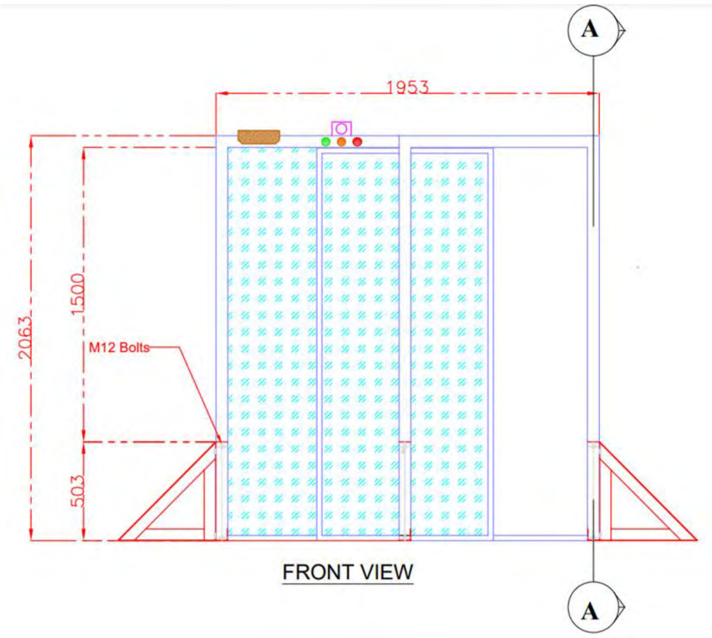

Figure 9: Sliding Door Design

ACKNOWLEDGMENT

The study was supported by the University of Zambia (UNZA). The dataset used for this study was collected from the Kaggle website. The author would like to thank the Computer Science Department (UNZA) for their input in the study. I would also like to acknowledge and thank my father Dr John Simwinga for all the support and guidance he had given me throughout this study.

REFERENCES

[1] J. Page, D. Hinshaw, and B. McKay, "In Hunt for Covid-19 Origin, Patient Zero Points to Second Wuhan Market – The man with the first confirmed infection of the new coronavirus told the WHO team that his parents had shopped there," The Wall Street Journal, 26-Feb-2021.

[2] Al Jazeera, "China reports First Death from mysterious outbreak in Wuhan," Coronavirus pandemic News | Al Jazeera, https://www.aljazeera.com/news/2020/1/11/china-reports-first-death-from-mysterious-outbreak-in-wuhan .

[3] "Who statement on novel coronavirus in Thailand," World Health Organization, https://www.who.int/news/item/13-01-2020-who-statement-on-novel-coronavirus-in-thailand.

[4] D. B. Taylor, "A timeline of the coronavirus pandemic," The New York Times, https://www.nytimes.com/article/coronavirus-timeline.html#:~:text=On%20Jan.,at%20the%20market%20in%20Wuhan .

[5] Novel coronavirus (2019-ncov) - world health organization (WHO), https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200121-sitrep-1-2019-ncov.pdf?sfvrsn.

[6] "China's total number of coronavirus deaths reaches 304 as of end-Feb 1 - state TV," Reuters, https://www.reuters.com/article/china-health-id INB9N28N00P.

[7] "Coronavirus: First death outside China recorded as total fatalities pass 300," The Guardian, https://www.theguardian.com/global-development/2020/feb/02/coronavirus-deaths-hong-kong-health-workers-to-strike-china-border-.

[8] "Africa identifies first case of coronavirus disease: Statement by the director of Africa CDC," Africa CDC, https://africacdc.org/news-item/africa-identifies-first-case-of-coronavirus-disease-statement-by-the-director-of-africa-cdc .

[9] "Coronavirus: First death confirmed in Europe," BBC News, https://www.bbc.com/news/world-europe-51514837.

[10] C. M. L. MELO, G. A. S. SILVA, A. R. S. MELO, and A. C. FREITAS, "Covid-19 pandemic outbreak: The Brazilian reality from the first case to the collapse of Health Services," Anais da Academia Brasileira de Ciências, vol. 92, no. 4, 2020. doi:10.1590/0001-3765202020200709\

[11] "Coronavirus covid-19 risk increased to 'very high' but containment still possible | UN news," United Nations, https://news.un.org/en/story/2020/02/1058331 .

[12] "Who director-general's opening remarks at the media briefing on COVID-19 - 11 march 2020," World Health Organization, https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020.

[13] "First 100 persons with Covid-19 - Zambia, March 18–April 28, 2020," Centers for Disease Control and Prevention, https://www.cdc.gov/mmwr/volumes/69/wr/mm6942a5.htm

[14] "Who coronavirus (COVID-19) dashboard," World Health Organization, https://covid19.who.int/.

[15] N. Cimolai, "In pursuit of the right tail for the COVID-19 incubation period," Public Health, vol. 194, pp. 149–155, 2021. doi:10.1016/j.puhe.2021.03.011

[16] G. Churiso, K. Diriba, H. Girma, and S. Tafere, "Clinical features and time to recovery of admitted COVID-19 cases at Dilla University Referral Hospital Treatment Center, South Ethiopia," Infection and Drug Resistance, vol. Volume 15, pp. 795–806, 2022. doi:10.2147/idr.s356606

[17] E. M. Mulenga, "Spread of covid-19 pandemic in Zambia: A mathematical model," Aquademia, vol. 4, no. 2, 2020. doi:10.29333/aquademia/8375

[18] P. N. Achyutha, S. Hebbale, and V. Vani, "Real time covid-19 facemask detection using Deep Learning," International journal of health sciences, pp. 1446–1462, 2022. doi: 10.53730/ijhs.v6ns4.6231

[19] L. Mucheleng'anga and C. Himwaze, "The role of forensic pathology in the COVID-19 pandemic in Zambia," Forensic Science International: Reports, vol.2, p.100147,2020. doi: 10.1016/j.fsir.2020.100147

[20] A. Nowrin, S. Afroz, Md. S. Rahman, I. Mahmud, and Y.-Z. Cho, "Comprehensive review on facemask detection techniques in the context of covid-19," IEEE Access, vol. 9, pp. 106839–106864, 2021. doi:10.1109/access.2021.3100070 2020.doi:10.1136/medethics-2020-106314

[21] D. H. Ng et al., "Feasibility and utility of facemask sampling in the detection of SARS-COV-2 during an ongoing pandemic," European Journal of Clinical Microbiology & Infectious Diseases, vol. 40, no. 12, pp. 2489–2496, 2021. doi:10.1007/s10096-021-04302-6

[22] V. Balasubramaniam, "Facemask detection algorithm on Covid community spread control using EfficientNet algorithm," June 2021, vol. 3, no. 2, pp. 110–122, 2021. doi:10.36548/jscp.2021.2.005

[23] G. Bonala, T. Kumar, and V. K. Gunjan, "Raspberry pi based crowd and facemask detection with email and message alert," Studies in Computational Intelligence, pp. 369–375, 2022. doi:10.1007/978-3-030-96634-8_35

[24] M. L. Mokeddem, M. Belahcene, and S. Bourennane, "Covid-19 risk reduce based YOLOV4-P6-facemask detector and DeepSORT Tracker," Multimedia Tools and Applications, vol. 82, no. 15, pp. 23569–23593, 2022. doi:10.1007/s11042-022-14251-7

[25] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, California, USA, 2019, pp. 6105-6114.

[26] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of Simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. doi:10.1109/cvpr.2001.990517