Face Mask Recognition Using Artificial Intelligence with Convolutional Neural Network for Access Control

Joseph Owusu [abc1], Reginald Nyarko [ab], Joshua Nti [ab], Adu Benedict [ab]

[a] Department of Electrical and Computer Engineering
[b] University of Energy and Natural Resources Sunyani, Ghana
[c] Autonomous University of Madrid, Spain
[1] joseph.owusu@uenr.edu.gh

Abstract— The outbreak of the Corona Virus pandemic has crippled a lot of nations and has affected the health and well-being of people all over the world. Since its inception, many measures and solutions have been put in place in a considerate effort to curb and reduce the spread of the highly contagious virus. A key protocol put in place is the mandatory wearing of face masks. This paper presents a face mask detection system for access control which can serve a plethora of purposes ubiquitously in many fields such as conferences, social gatherings, high schools, universities, and many more. The face mask recognition system in this paper is built using the image classification method: Convolutional Neural Network. To achieve the working model of the system, the various procedures to be carried out are, collecting of image data, preprocessing of image data, splitting of image data into training and validation data, validation of the model, and implementation and deployment of the model on the raspberry pi 4 embedded system. After the model is deployed, a door access system is controlled based on an algorithm on the system. This will go a long way to reduce the spread of the virus which has harmed so many.

Index Terms—Artificial intelligence, COVID-19, image classification, Machine learning, neural networks.

I. INTRODUCTION

The field of Artificial Intelligence (AI) has seen exponential development and application in various sectors in recent years, dating from the 1900s. Artificial Intelligence is employed in many parts of our lives today. Machines and systems are now equipped with the ability to use memory, knowledge, experience, understanding, reasoning, and judgment to tackle issues and adjust to new circumstances. As defined subjectively by Alan Turing during the Turing Test, "A computer would deserve to be called intelligent if it could deceive a human into believing that it was human" (Stuart et at., 2009; Wolfgang et al., 2017). The subsets of AI are machine learning and that of machine learning is deep learning. There are four main machine learning methods, namely: Supervised Learning, Unsupervised Learning, Semi-supervised Learning, and Reinforcement Learning. Deep learning is a supervised machine learning method that employs the science of learning from obtained data using structured algorithms to create artificial neural networks which closely emulate and bio-mimic how neurons work in humans. AI implementation on embedded systems encapsulates making an AI algorithm or model run seamlessly on embedded devices. In recent years this technology has been more and more widely adopted, from simple appliances like autonomous vacuum cleaners to autonomous vehicles which might soon become normal (Zheng, 2020).

It is easy to see how tasks that involve pattern recognition (PR) would be a natural candidate for use in embedded systems since these systems interact closely with the environment around them. Most PR applications involve perception through input devices or sensors, understanding of its environment, and performing an actuation or displaying information based on this (Perez-Cortes et al., 2009). In (Nasir et al., 2019), the Arduino microcontroller-based platform was used for the control and actuation of the door access system and a webcam was used for the image capture. Arduino is an electronic circuit board that is open source and uses a microcontroller chip as its core component. It can be reprogrammed using the Arduino integrated development environment (IDE) application and C/C++ programming language. The Arduino microcontroller was connected to a computer that has a special application that was designed with the C# programming language and MySQL server as a database. The application that was built is a face mask recognition mechanism that was built using OpenCV. From the results obtained, it was observed that the system had a good accuracy but a very slow response. This is due to the relatively slow graphical processing power and the sampling rate of the Atmega328 P microprocessor and webcam. Also, there was the implementation of Principal Component Analysis (PCA), a type of traditional machine Learning Algorithm in masked and non-masked facial recognition systems in (Ejaz et at., 2019). Principal Component Analysis is a machine learning approach that is used in facial recognition applications to extract features. It's a method for lowering the dimensionality of such datasets while maintaining usability and minimizing information loss. It accomplishes this by generating new uncorrelated variables that sequentially optimize variance. From the results obtained in in (Ejaz et at., 2019), traditional machine learning algorithms like the Principal Component Analysis yield less accurate results as compared to convolutional neural networks for faces without a mask than faces with a mask. The PIC microcontroller was also employed in (Naing et al., 2019) which poses a huge limitation as compared to faster computing platforms like the raspberry pi due to low clock speed, and a smaller number of input/output pins.

A major application of AI in the field of computer vision deals with computers and our devices gaining a high-level understanding of the things captured on camera and emulating automated tasks the human visual system does (Huang et al., 1996). In (Sanjaya et al., 2020), the authors combine computer vision techniques with a strong backbone of an Artificial Intelligence model known as a convolutional neural network (CNN). Convolutional neural networks are a type of deep neural network that is most typically used in image processing. As CNN can be extremely complex and resource-intensive for mobile and embedded system applications, in Sandler et al., 2018) a neural network architecture called mobilenetV2 is introduced tailored for these constrained environments. MobileNetV2 is a lightweight convolutional network architecture that seeks to perform well on mobile and embedded applications. This was used in (Sanjaya et al., 2019), the drawback in this design was, the system was not physically modelled for the real-world deployment and usage for optimization. Also, according to the anti-theft system in (Jahnavi et al., 2019), the principal component analysis was also deployed using the matrix laboratory (MATLAB) environment. The major drawback offered by that system is that the facial capture computations, preprocessing functions like resizing, sharpening, cropping, normalizing, and enhancement, as well as facial detection functions, are all performed on a laptop computer unlike a cost-effective miniature microprocessor-based platform like the raspberry pi 4. This limits the versatility and ease of implementation of the project in the real world due to its robustness. The system built in this paper combines all the advantages of versatility, ease of implementation, speed and response as well as accuracy that are not combined advantages offered by the papers discuss above in the literature.

Looking at the prevailing worldwide pandemic and the measures to be put in place to ensure safety, wearing a face mask has become key to preventing the spread of the virus. Heighten rate of infections of coronavirus spread in Ghana and other parts of the world have been related to negligence of protocols put in place to prevent the spread of covid-19 such as wearing of a mask. This paper aims to design and build an access control system automated by an electronic lock in which its controls are dependent on whether a person accessing the door lock system is wearing a facemask or not. The objectives are to design and train machine learning models to be able to classifier faces based on whether or not an individual wears a face mask and deploy a machine-learning model on a Raspberry Pi computer. Also, a prototype model is built to demonstrate a lock system controlled by image model prediction.

II. DESIGN AND IMPLEMENTATION

Deep learning has caused great improvements in the field of artificial intelligence, mostly in pattern and object recognition and also image processing in recent years. Current neural network algorithms and techniques have achieved amazing performance improvements, such that even lightweight neural networks applied in embedded devices can work seamlessly (Wu et al., 2019). The deep learning algorithm used in this project is MobileNetV2 which is a convolutional neural networks (CNN) technique developed by google for increased performance for mobile and embedded applications. The steps involved in the study include the training process for the model, the deployment of the model on the raspberry pi computer, hardware connections, and the implementation of the model with the connected devices.

A. TRAINING THE MODEL

The machine learning model is trained using several steps to ensure model predictions are highly accurate, these steps are illustrated in figure 1. The first step used in training is data collection.

Fig. 1. Steps in building model

The dataset used is from the face mask detection dataset by Ashish Jangra on Kaggle (Jangra, 2021). There was a total of 2000 images used, 1,000 images with masks, and another 1,000 images without masks. The data was labeled in two groups, with masks, and without masks. The pre-processing of data is the next step, in this step, the images are resized and converted into arrays. The one hot encoded preprocessing algorithm is then applied to it. Resizing is crucial to image pre-processing since the model will run better with smaller images. Nevertheless, the image should not be too small that crucial features are lost. The images used in this project are resized to 224 x 224 pixels. The images are then stored into numerical python (NumPy) arrays with 32 floating-point numbers. The strides of array inform the base model of the number of bytes that can be stored in each column. Since precision in recognition of the images, a float32 data type was used. One hot encoding was used to create a binary column corresponding to each category given each unique value in each category a feature. By default, the data type of the encoder is a numerical floating point, allowing the algorithm to identify the labels assigned. The procedure ensures that the collected data is split for analysis, training, and testing. For model training, 80% of the data are used, and 20% for testing. The data is shuffled using a random state. This controls the shuffling of the data before data is split. This is done to ensure the validity of the model after training to prevent the same sequence of data each time the model is run.

The model is then built using CNN. It is necessary to provide a head model to the base model in the usage of CNN. The head model is used to prepare and tune the model using its hyperparameters. The hyperparameters used in the head model are Average Pooling, Flatten, Dens and Dropout. The head model is set to receive the base model and set to output. This makes the head model an output function that takes the base model as input, prepares it, and outputs it after all hyperparameters have been applied to it. One problem with the output function mappings is that the functionality in the input is sensitive to its position. A proposed solution to the output function mappings problem is to reduce the sample of a particular feature to correct this sensitivity. This makes the resulting sampled function mappings more resilient when the feature position changes in the image (www.machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/, 2021). To solve this average pooling of 7 by 7 pixels is used as the CNN may be repeated more times in the model. Average pooling takes the average of the pixels in a specified dimension range as a square matrix and averages them. This makes the sharper features blend with other images for better recognition. The convoluted sum will need to be flattened. Flattening of the layer reduces the spatial dimension to generate a sequential input. A dense layer, which constitutes the neurons of the CNN that are connected forming a layer of 128 nodes. This is done so that the data can be inserted into the neural network. Hyperparameters of CNN specify how the neural network will be trained. One hypermeter of CNN is the dropout rate. This specifies how much of the nodes in the CNN will be used for training. A dropout of 0.5 is used to cater for the overflow during flattening. This will mean that there is a 50% probability that the nodes for a layer will be dropped as the model is being trained. This is to prevent overfitting of the model which will improve the accuracy.

Another dense layer is added as the last set of the convolution architecture with softmax as its activation. The dense layer requires activation as a parameter to convert the normal function with its dense layer to an artificial neuron that will enable the CNN to learn from the input the dense layer receives and the output it provides. Softmax was utilized as the activation for the layer of the classification network since the result will be expressed as a probability distribution. The head model is then added on top of the base model and this is the final model to be trained. A loop is made to run on all the base model layers and they are frozen to prevent them from being updated in the first training process. The model is compiled using Adaptive Optimizer Estimator (ADAM). ADAM was used to optimize the model which maintains a single learning rate for all weight updates. The learning rate was 0.0001 and decays over five seconds (5s). The system metrics were set to accuracy to present the level of accuracy as the system process runs its iteration. To find the steps per epoch we divided the length of the training data by the model batch size. The batch size is set to 32. Validation steps are also stepped by determining the length of the test data by the batch size. The model was then trained at 20 epochs and saved as an h5 file. The model is tested by predicting whether or not the test image has a person wearing a mask or not.

B. MODEL DEPLOYMENT AND IMPLEMENTATION

After the model is trained, it is then used as a function on the raspberry pi. The program takes a real-time image with its attached camera and processes the captured image on basis of whether or not a person is wearing a face mask rightly and then apply the results of the processing in the decision of whether to open a lock. The lock access is not the only application as the information could be used in general decision making such as access control and counting, and capacity control. The other hardware components of the system consist of a solenoid lock, relay, buzzer, a raspberry pi, 8 Mega Pixels (MP) camera, and a 12 Volt DC adapter.

C. SYSTEM CONFIGURATION

The system block diagram in figure 2 shows how the various parts of the system interact. The raspberry pi model used is a high-performance processor which uses Raspbian, a Linux Operating system. It serves as the brain of the system processing the video received from the pi camera which is a camera specifically designed for the raspberry pi and sending a signal to the buzzer or relay base on the saved model prediction. The relay in the system is an electromagnetic switch that controls the lock. A relay is used since the solenoid valve used operates on 12V DC and the raspberry pi cannot provide voltages that high. The relay switch takes a 5V input from the raspberry and then allows a 12V DC signal to flow to the lock.

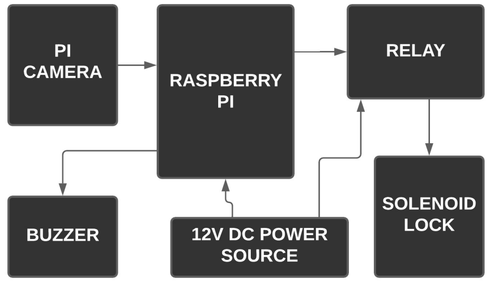
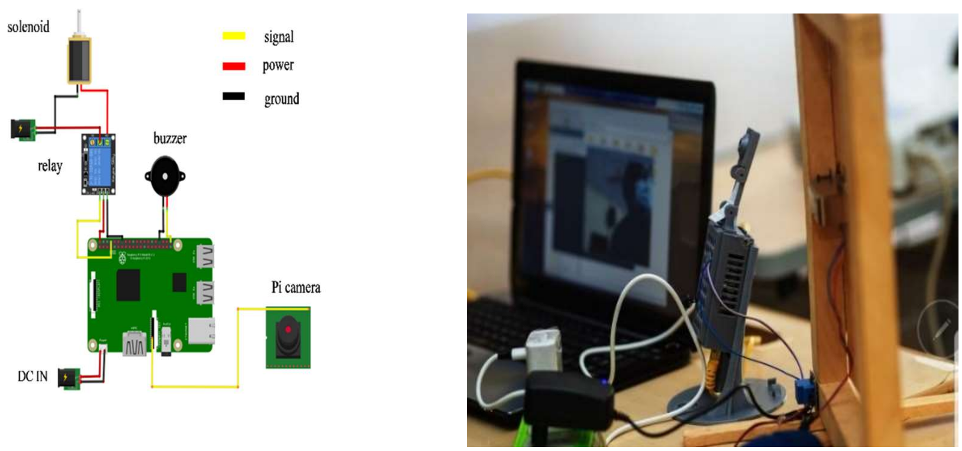


Fig. 2. Block Diagram

The system also uses a 12V adapter as the power source since it can power both the processor and solenoid lock. Figure 3 (a) shows the exact circuitry of the system and figure 3(b) shows the built and implemented system.



(a) (b)

Fig. 3. (a) System Diagram (b) The Proposed Prototype System

D. PROCESSES AND IMPLEMENTATION

The model is implemented using OpenCV, a programming library with functions directed toward computer vision applications in real-time. Several processes happen in less than a second to determine whether or not to open the lock. First, the video stream from the camera is read frame to frame, the system checks each frame for the presence of a face and if the face is detected, the system proceeds to the next process to check for a face mask. The workflow of the system is shown in figure 4. From each frame obtained, the image is prepared for prediction by resizing the

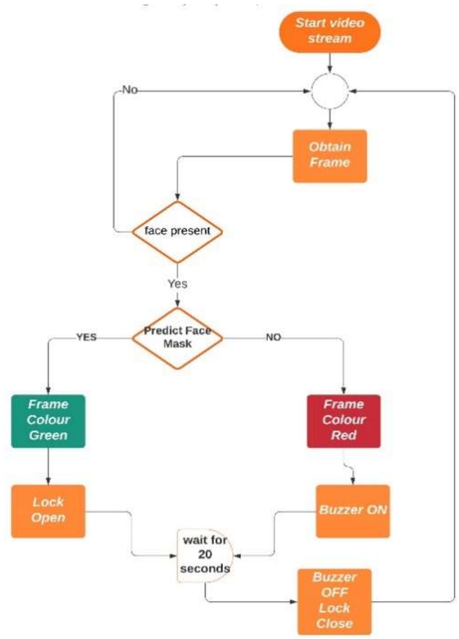
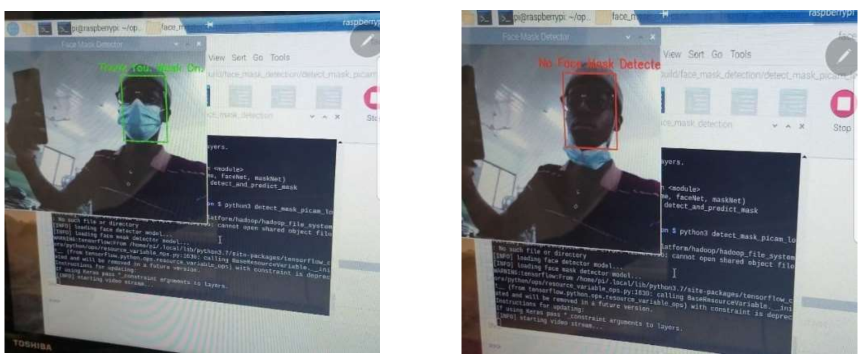


Fig. 4. Flowchart

image and converting it to an array. The prediction is then done using the trained model. Model prediction decides whether or not a person with the frame is wearing or is not wearing a face mask. Based on this prediction the system decides to either give access to individuals wearing a mask or deny access to people who do not wear a mask, letting the noise from the buzzer if the individual is without one. Shown in figure 5 (a) and (b) is the system reaction when an individual is with or without a mask.



(a) (b)

Fig. 5. (a) When the face mask is on (b) When the face mask is off

The paper aims to contribute to bolstering Agenda 2063, the Africa we want. The agenda 2063 is a strategically implemented long-term vision that seeks to drive Africa to be integrated, prosperous, self-driven independent from other continents to ensure that Africa becomes a dominant force to compete with other continents globally. The flagship projects of Agenda 2023 include ICT resolution, education, and health (Agenda 2063, 2019; Flagship Projects of Agenda 2063, 2020). This paper helps to realize the application of technology in education to propose a solution to probing the health crisis of the Covid-19 virus. The implementation of machine learning in curbing the ravaging coronavirus among member states of Africa is a firm step in transforming Africa into a healthy e-Society. Development of the face mask recognition system for access control will be an innovation that can provide a transformative service un-parallel to existing ways of controlling the spread of the coronavirus indoors, that is, a security agent checking for the wearing of nose mask before access is granted.

III. CONCLUSION

In conclusion after data collection, training, validation testing, deployment, and testing of the system, this paper presents a solution that uses machine learning and image processing for facial mask recognition and access control in the effort to curb the spread of the COVID-19 virus. Considering the level of spread of the Covid-19 virus and how long it might take to get rid of the virus, it is necessary to employ current technologies that will allow us to best minimize the damage such as automation of the nose mask wearing detection process. Applications of the system will include use in public gatherings, hotels, and other institutions to ensure better observation of COVID-19 mask wearing protocols.

IV. REFERENCES

(1) Stuart J. Russel and Peter Norvig (2009) Artificial Intelligence In: Artificial Intelligence: A Modern Approach, 3rd edition. Pearson Education Inc.

(2) Wolfgang Ertel (2017) Introduction In: Introduction to Artificial Intelligence, 2nd edition. Springer International Publishing.

(3) J. X. Zheng, 2020. Invited Talk: Software Engineering, AI and autonomous vehicles: Security assurance, 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 1-1, doi: 10.1109/PerComWorkshops48775.2020.9156201.

(4) J. Perez-Cortes, J. Guardiola and A. Perez-Jimenez, 2009. Pattern Recognition with Embedded Systems Technology: A Survey, 20th International Workshop on Database and Expert Systems Application, pp. 19.

(5) Jaunuardi Nasir, Azizul Azhar Ramli and Micheal, 2019. Design of Door Security System Based on Face Recognition with Arduino, International Journal on Informatics Visulization, ISSN: 2549-9610.

(6) M. S. Ejaz, M. R. Islam, M. Sifatullah and A. Sarker, 2019. Implementation of Principal Component Analysis on Masked and Non-masked Face Recognition, 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1-5.

(7) San San Naing, Thiri Oo Kywe, Ni Ni San Hlaing, 2019. Face Recognition Based Intelligent Door Control System, International Journal of Trend in Scientific Research and Development, pp. 643-648.

(8) T.S. Huang, 1996. Computer Vision: Evolution and Promise, 19th CERN School of Computing. Geneva: CERN. pp. 21–25. doi:10.5170/CERN-1996-008.21. ISBN 978-9290830955.

(9) S. A. Sanjaya and S. Adi Rakhmawan, 2020. Face Mask Detection Using MobileNetV2 in the Era of COVID-19 Pandemic, 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), pp. 1-5.

(10) M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

(11) S. Jahnavi and C. Nandini, 2019. Smart Anti-Theft Door locking System, 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), pp. 205-208.

(12) Hao Wu, Qi Liu and Xiaodong Liu, 2019. A Review on Deep Learnning Approaches to Image Classification and Object Segmentation, Computers, Materials and Continua 58(2), pp. 575-597.

(13) A. Jangra, Face Mask Detection ~12K Images Dataset. Accessed on 26/5/2021, https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset.

(14) A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. Accessed on 26/5/2021, https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/.

(15) Agenda 2063: The Africa We Want, African Union, 1 Jan. 2019, au.int/en/agenda2063/overview.

(16) Flagship Projects of Agenda 2063, African Union, 2 Dec. 2020, au.int/en/agenda2063/flagship-projects.