Chapter 10
Face Mask Detection Based Entry Control Using XAI and IoT

Yash Shringare, Anshul Sarnayak, and Rashmi Deshmukh

Abstract Today it has become mandatory for all the citizens to wear a face mask to protect them from COVID-19. Also taking two doses of vaccine is a must to visiting public places and currently, the only method to verify whether a person is fully vaccinated is by showing a vaccine certificate. The proposed application is helpful for elderly people who find it difficult to use smart phones. The shop owners, offices, banks, or any public place can check for restrictions of entry if anyone is not wearing a mask. As a result, no need for any guard to keep an eye on people. Machine learning techniques with Explainable AI (XAI) can solve these problems easily and results are made understandable to end-users because of the explaining ability and interpretability of neural network models. The system performs well for prediction and gives more accurate and trustworthy predictions. Hence XAI is more reliable in healthcare systems. The proposed system is implemented completely on Raspberry Pi allowing a complete embedded application. The application is developed using Python and HTML. PyCharm/Visual Studio Code with the help of an open-source library is used for training, defining, etc. Machine learning models used for the system are Tensorflow.js, Keras, OpenCV, etc. The whole application can run on a microcontroller such as Raspberry Pi, which allows one to simply plug and play the system at any time.

Keywords CNN · XAI · OpenCV · Mask detection · Raspberry Pi · Bootstrap · HOG

Y. Shringare · A. Sarnayak
Department of Technology, Shivaji University, Kolhapur, Maharashtra, India

R. Deshmukh (✉)
Department of Technology, Shivaji University, Kolhapur, India
e-mail: rvm_tech@unishivaji.ac.in

M. Mehta et al. (eds.), *Explainable AI: Foundations, Methodologies and Applications*, Intelligent Systems Reference Library 232,
https://doi.org/10.1007/978-3-031-12807-3_10

10.1 Introduction

Emerging studies show that wearing a mask can protect us from the spread of COVID-19. Multi-layered masks have good performance in blocking exhaled small droplets. Hence no one without a mask is allowed to enter the mall. The proposed model for face mask detection can solve this type of problem. The proposed model not only detects a face mask but also controls entry points in any mall, Bank, or cinema hall. A neural network with explainability helps to improve the performance by detecting the system well. It finds where the model fails. Due to its interpretability, XAI is used in the application of image processing (Wells and Bednarz 2021). Also, explanations of the prediction made by the neural network model make them more reliable and trustworthy. XAI plays an important role to solve this issue with the help of CNN (Samek et al. 2019).

The problem is the traditional way of ensuring the mask is done by individual check. This is difficult for people especially old people who don't have a smartphone to always show their vaccine certificate. Also, this problem has negatively impacted business owners because it is hectic work to constantly remind people to wear a mask.

Proposed system with web-based application is designed to collect the vaccine certificate and Government ID. Then verification is done using AI and a link is provided to register face encodings also known as embeddings are taken through the web application and stored in a central database using pickle these encodings can be accessed from any scripts. Embeddings are imported to Raspberry Pi from the database RPi uses the Pi Camera and then prompts the user for facial scanning and checks whether the person is already registered on the web portal if successful prompts the user for wearing a mask and upon successful detection of mask send a command through GPIO's to 12 V DC Solenoid lock to open the door.

As per ruralindiaonline.org, there are 137.9 million elderly people (people above 60) living in India and these people often have problems using smartphones, and finding certificates on smartphones proves to be a difficult task. Our application is designed to help such people through our web application and Raspberry Pi. This application is a deep learning model using CNN. XAI provides more visual interpretability than CNN which uses a separate filter to represent the specific object. XAI builds a black box mode first and simultaneously gives a post hoc explanation of how CNN classifies the images and also gives verification of the method used for classification. Hence with XAI, it is possible to design and develop a neural network model which removes bias and presents better predictive performance (Oh et al. 2021). Further, it is integrated and deployed using the library Tensorflow. The web application for this proposed model uses the face-recognition library which is built upon dlib a C++ toolkit.

10.2 Literature Review

Radzi et al. use Convolutional Neural Network (CNN) technique consists of eight layers. Purposed IoT-based face recognition Home security system uses Raspberry Pi. To capture some additional information and to improve accuracy and processing time of the given system had used various activation functions. System is designed with open source software Python and Keras. System uses hundred epocs at the start and for testing twenty epocs per iterations are executed. For IoT Blynk app was used. Blynk is an app that is used to control Raspberry pi and many other microcontrollers. Simply drag and drop the widgets in the app giving a notification to the owner whenever the doorbell was pressed and giving option to open the lock (Syafeeza Ahmad Radzi et al. 2020).

Another work related to this field was by Amritha Nag et al. (2018) in creating system for controlling door access with the use of IOT and face recongition system. It uses OpenCV based face recognition system using Haar classifiers for a face. The pi camera is employed to capture the image. The main aim of this system was to implement entire processing on the Pi computer and sensors for detection of emotions of the face. GSM module is used by installing Subscriber Identification Module (SIM). To control the lock mechanism. This paper describes the use of the Viola-Jones Face Detection Method in three steps. It is implemented with first evaluating the features of image with Haar function and some of the most relevant features are selected for classification with the use of Adaboost algorithm. Further cascaded classifier is used to give the door access (https://docs.opencv.org/).

Adam et al. portrayed a method for advanced facial recognition which firstly involved the detection of faces using HOG (Histogram of Oriented Gradients) for each pixel. Here surrounding pixels are found and gradient painting is done by drawing an arrow that points to a darker region. A further similar image was found which is matching mostly with the HOG pattern. Paper proves the effectiveness of the Jones classifier for matching images with the face landmark estimation algorithm. System uses ML dlib library which detects face on the basis of sixty eight different points on the face which will give uniqueness to face. These specific points are given as input parameters to CNN. After having measurements for each faces to compare them Adam used a SVM classifier. If measurements were similar the persons in the two images were marked identical (Geitgey 2017).

Detection of different poses and emotions of the face is done by open source platform of python as TensorFlow tools and libraries (Tensorflow 2019). Smaller DNN like MobileNet are used to clarify facial images (Friedhoff and Alvarado 2018).

System represents a real time detector for face masks. It uses MobileNetV2 for classification of facial images (Sanjaya and Adi Rakhmawan 2020).

More et al. Uses face recognition for monitoring the attendance (More et al. 2021). System gives attendance analysis for teaching staff.

Anand et al. demonstrated Face Recognition based Attendance system. System uses controlling door access with various DNN tools. Different tools are used by the

system like small and chip computer Raspberry Pi, 360 degree scanner RPLIDAR A1 and deep learning USB Movidius NCS2 etc. (Anand 2021).

Another research improves the performance of the system with the use of RGB image data for avoiding false positive results and attcks of face spoofing. It gives very high accuracy and F1 score for recognition of facial images (Ko et al. 2021).

10.3 Methodology

Following section gives explanation of methodology used by the proposed system (Fig. 10.1).

The integration of the webcam with the website is the first stage in the flow diagram. Further user registers on the web app by providing their Vaccine Certificates and Government ID. Then the user has to provide face encodings which are stored in a database. This completes the use of the web application module. For training in face mask detection, two classes of images were taken with mask and without the mask. Images are pre-processed and then given as input for the CNN model which is implemented using TensorFlow. After the training process, the trained model has loaded into a script this script. The further script imports the face encodings from the web app and scans the video feed from the camera. If the face is found in the database it prompts the user to wear a mask and sends commands to Raspberry Pi's web server to open the door using GPIO.

10.3.1 Web Application Execution

The backend used for this web application is Django and front end tools used are HTML, JavaScript, bootstrap, and CSS etc. Web application collects the vaccine certificate and Government ID. It provides a link to register face encodings and saves them in the database using pickle.

Fig. 10.1 Flow diagram

ML model is implemented with Django.ai ML framework. It gives a set of tools and libraries to develop ML project with ease.

Pickel: A data structure used by complex object hierarchies is serialized when it is converted into character stream with serialization process and transferred across the network. This is done by Pickel module of Python.

10.3.2 Implementation

This contains detail explanation of the different steps used for implementation.

First step is collection of image data for face recognition; further model is trained and tested with various activation functions.

i. **Detection of facial image and data collection**

Face detection is performed based on the following 4 parameters.

a. Haar Features

Square shape function is scaled with the use of Haar function. Considering all the variations in size, and position of all these features, we have calculated about 160,000+ features. This involves a 24 × 24 window consisting of many redundant features and many of them are not useful.

b. Integral Image

Internal image contain each pixel with summarized form of pixels present to left and above.

c. Adaboost

Adaboost eliminates all the redundant features and narrows them down to several thousand very useful features. For final classification weighted combination of selected best feature is done.

d. Cascading

Let us have an input image of 640 × 480 pixels resolution; we need to move this 24 × 24 window all through the image. We must analyze 2500 features obtained by Adaboost for each 24 × 24 window. Further, a linear combination of all those 2500 outputs is used to see whether it exceeds a given threshold. Finally, a decision is taken whether a face is detected or not. Instead of applying 2500 features on every single 24 × 24 window at a time, we use cascades.

We have defined two class paths as with mask and without the mask. Then each frame captured by the camera is read and faces are detected in the capture frame by using the above cascade classifiers. A further rectangle is drawn around the face to crop that part and the images are saved in respective folders (Fig. 10.2).

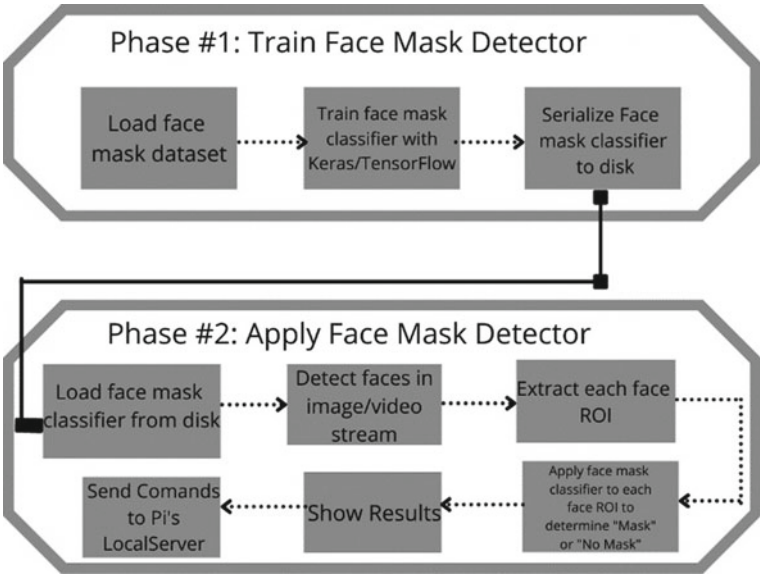

Fig. 10.2 Train and apply

ii. Model Training

Model Training was done using Tensor flow. Some layer modules from the Tensor flow library were used in the proposed system.

Dense: It gives fully connected NN layer where each neuron gets output from previous layer and further passes one output to next layer.

MaxPooling2D: MaxPooling2D is useful for down sampling the input by taking the maximum value for each channel of the input over an input window (Fig. 10.3).

Flatten: To flatten multi-dimensional data into one-dimensional data, utilize the Flatten layer.

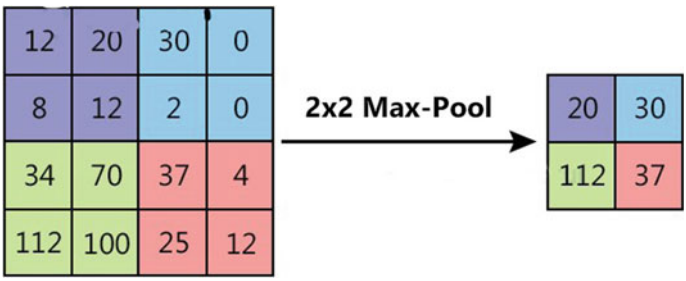

Fig. 10.3 Down sampling of input image

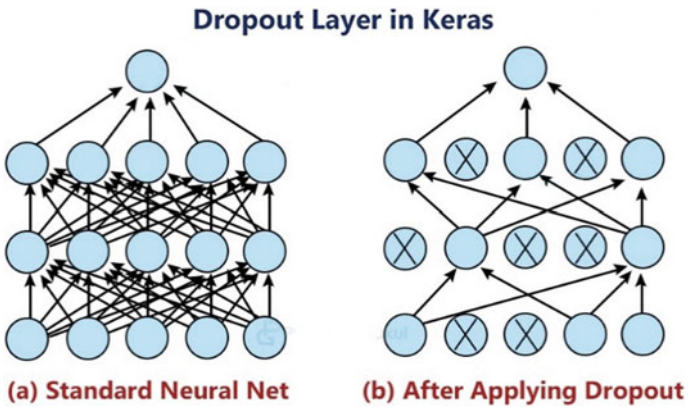

Fig. 10.4 Droupout layer

Dropout: Dropout is a technique for preventing over fitting in neural networks by disregarding neurons at random during the training phase (Fig. 10.4).

i. **Data Preprocessing**

In the data preprocessing; part images from both directories (class index) with_mask and without mask were collected and stored in list x (for image data) and y (for class index) respectively. Resize function from OpenCV is used to resize image to 100 × 100 sizes.

Resized image array is divided by '255' sothat deep learing performs well. Further all the lists were converted to NumPy array as the model feeds on more number of images simultaneously. Generally this process is known as a batch. Further reshaping of the array is needed.

To create model architecture sequential API is used. This plain stack of layers contains each layer with one input and one output. Model. Add () function adds a layer to the model this Model architecture contains four Convolution layers and three Dense Layers. A batch of arrays is fed into the first convolution layer. 'input shape' determines the shape of the input array. Final layer has two output so data is classified into two different classes. 'Softmax' is the last layer's activation function as the categorical data was used. Finally, the model was compiled, and the loss function "sparse_categorical_crossentropy" is used.

10.3.3 Activation Functions

Activation functions comprise mathematical formulas for finding the output of a neural network in Deep learning models. Each neuron in the network has an Activation Function that determines whether or not to stimulate that neuron. The relevance of each neuron's input to the model's prediction is used to make this decision.

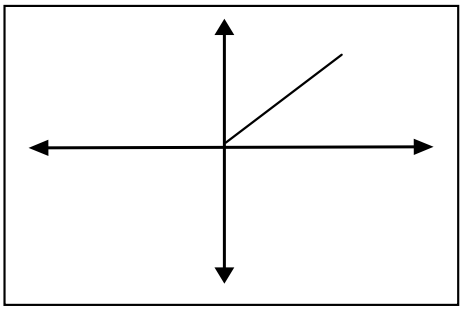

Fig. 10.5 Graph of RELU activation function

The activation functions employed in the proposed model are listed below:

i. **Softmax Activation Function**

The Softmax function is frequently used to solve classification problems involving several classes. It's used to convert the output of neural networks into a number between 0 and 1. It's utilized to signify a certain level of "probability" in the network's output. As we are dealing with probabilities, the Softmax function's scores will all add up to 1.

- Formula of Softmax Activation Function:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{10.1}$$

ii. RELU Activation Function (Rectified Linear Unit):

The RELU layer finds and eliminates any negative values from the filtered image and replaces them with zeros while leaving only positive values. To keep the values from aggregating to zero, RELU is utilized (Fig. 10.5).

$$f(x) = \begin{cases} 0 \; if \; x < 0 \\ x \; if \; x \geq 0 \end{cases} \tag{10.2}$$

10.3.4 Raspberry Pi Webserver

The web server is implemented on Raspberry Pi HTML page was developed to control GPIO over the internet. If a mask is detected the above model will execute a command on Pi's HTML page using Selenium Web Driver. Admin connected to the local LAN network will also be able to override the system to control the GPIO (Fig. 10.6).

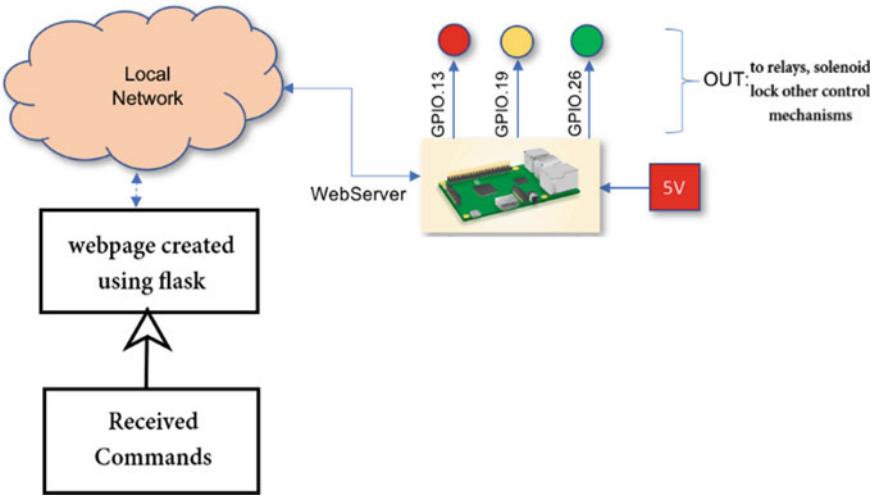

Fig. 10.6 Flow diagram of Pi webserver

10.4 Results

10.4.1 Dataset

We have created our database for the training. One with the mask as shown in Fig. 10.7 and one without the mask shown in Fig. 10.8

First, images for the mask class were captured and stored in the class_path as a mask. Then get_detection (frame) function. İs used to detect a face. Then img[y:y + h, x:x + w] extracts some of the portion of image and cv2.imwrite() used for storing the image details.

The count was set to "count >=500" so 500 images were collected and the loop closed automatically. After collecting with the mask shown in Fig. 10.8. We have collected data without mask and with mask in Fig. 10.7.

10.4.2 Model Summary

The model was built using layers API: Sequential model which is a linear stack of layers made by passing a list of layers to the add() function. Validation is one of the key advantages of working with a LayersModel as it compels to provide the input shape, which will be further used to validate the input.

Figure 10.9 gives a summary of model.summary() function. This includes, all layers in the model with name and type, each layer's output form, each layer's

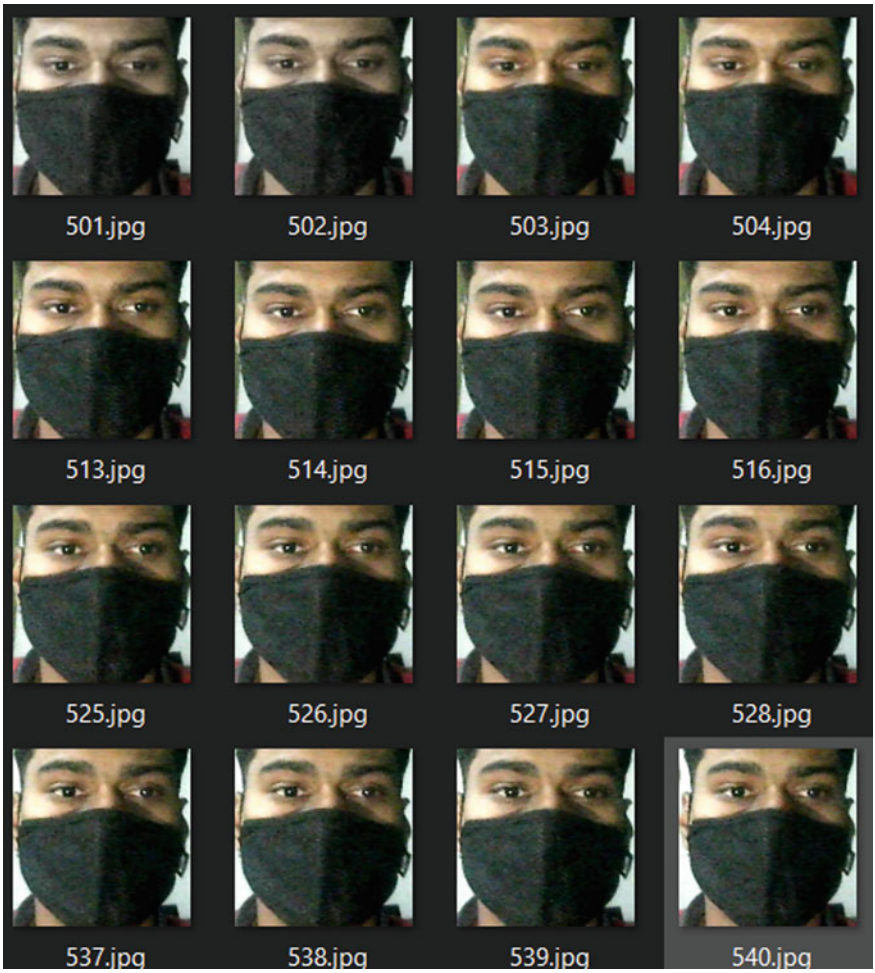

Fig. 10.7 Dataset with mask

weight parameters, generic topology of model and model's total number of trainable and untrainable parameters.

10.4.3 Model Evaluation

When we run the above statement we get: Output: 0.0282 loss–0.9950 accuracy. In the test dataset, we can observe that the model has an accuracy of above 99%. This shows a test accuracy of 99%, which should be acceptable. What it means to us that in 1% of the cases, the mask or without mask would not be classified correctly.

Fig. 10.8 Dataset without mask

10.5 Conclusion

Due to COVID-19, most malls and public places need to check a vaccine certificate and mask of each individual at the entrance. It becomes difficult for elderly people to find their certificates on mobile phones and many of them don't even have smart phones. The proposed application helps such people by using facial recognition. With the proposed system, we can detect the mask on a person's face and allow entry. Deep learning and machine learning models are used for image classification. In many cases, DNN results are not easily known to the end-user. But more understandable

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 98, 98, 64)        1792
_________________________________________________________________
activation (Activation)      (None, 98, 98, 64)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 49, 49, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 47, 47, 256)       147712
_________________________________________________________________
activation_1 (Activation)    (None, 47, 47, 256)       0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 256)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 21, 21, 128)       295040
_________________________________________________________________
activation_2 (Activation)    (None, 21, 21, 128)       0
_________________________________________________________________
dropout (Dropout)            (None, 21, 21, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 19, 19, 32)        36896
_________________________________________________________________
activation_3 (Activation)    (None, 19, 19, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 9, 9, 32)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 9, 9, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 2592)              0
_________________________________________________________________
dense (Dense)                (None, 100)               259300
_________________________________________________________________
dense_1 (Dense)              (None, 16)                1616
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 34
_________________________________________________________________
activation_4 (Activation)    (None, 2)                 0
=================================================================
Total params: 742,390
Trainable params: 742,390
Non-trainable params: 0
```

Fig. 10.9 Model.summary()

results are achieved with explainable AI (XAI) which will give the most trustworthy and interpretable results.

Most of the XAI uses RNN and the classification of an image is done with CNN. With the use of XAI, the CNN model results in an output image that shows the most relevant features needed for prediction. Local interpretable model- Agnostic Explanation algorithm is used with CNN to give explanations of prediction used by any classifying model. Explanations are given in the form of feature relevance or contribution to the prediction of a certain sample dataset. CNN Algorithm used in the proposed system is divided into four stages Haar Features Selection, Integral Images, AdaBoost and Cascading Classifier. The project is the epitome of IoT with Raspberry pi. Experimental results show that the proposed system with CNN gives more accurate and reliable results for image detection and classification. This Application has many advantages over the traditional way of enforcing strict lockdown rules and regulations in a country like India where the population is highly dense. The proposed application will provide an easy way of COVID prevention. With the use of XAI, it is possible to present more interpretable results of face mask detection and classification.

References

Anand, P.V.: Facial-attendance-on-Pi-with-LIDAR. (2021). https://github.com/AdroitAnandAI/FacialAttendance-on-Pi-with-LIDAR

Friedhoff, J., Alvarado, I.: Move mirror: an AI experiment with pose estimation in the browser using tensorflow.js. TensorFlow Medium. (2018). https://medium.com/tensorflow/move-mirror-an-ai-experiment-with-pose-estimation-in-the-browser-using-tensorflow-js-2f7b769f9b23/

Geitgey, A.: Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning. (2017). https://github.com/ageitgey/face_recognition

Introduction to OpenCV. https://www.geeksforgeeks.org/introduction-to-opencv/, https://docs.opencv.org/

Ko, K., Gwak, H., Thoummala, N., Kwon, H., Kim, S.H.: SqueezeFace: integrative face recognition methods with LiDAR sensors. J. SensS. **2021** (2021). https://doi.org/10.1155/2021/4312245

More, S., Kadam, N., Savla, S., Shelar, S., Shah, A., Mali, S.: CaptureIt!-A web-based attendance system using face recognition. In: IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), pp. 91–96. (2021). https://doi.org/10.1109/RTSI50628.2021.9597350

Nag, A., Nikhilendra, J.N., Kalmath, M.: IOT based door access control using face recognition. In: 3rd International Conference for Convergence in Technology (I2CT), pp. 1–3. (2018). https://doi.org/10.1109/I2CT.2018.8529749

Oh, J.H., Choi, W., Ko, E., Kang, M.: PathCNN: interpretable convolutional neural networks for survival prediction and pathway analysis applied to glioblastoma. Bioinformatics, **37**(1), i443–i450 (2021). https://doi.org/10.1093/bioinformatics/btab285

Syafeeza Ahmad Radzi, M.K., Mohd Fitri Alif, Y., Nursyifaa Athirah, Jaafar, A.S.: IoT based facial recognition door access control home security system using raspberry pi. Int. J. Power Electron. Drive Syst. **11**(1), 417 (2020). https://doi.org/10.11591/ijpeds.v11.i1.pp417-424

Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K.: Explainable AI: interpreting, explaining and visualizing deep learning. In: Lecture Notes in Computer Science book series (LNCS), vol. 11700. (2019). https://doi.org/10.1007/978-3-030-28954-6. ISBN 978-3-030-28953-9

Sanjaya, S.A., Adi Rakhmawan, S.: Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic. In: International Conference on Data Analytics for Business and Industry: Way towards a Sustainable Economy (ICDABI), pp. 1–5. https://doi.org/10.1109/ICDABI51230.2020.9325631

Tensorflow, J.S: Machine learning for the web and beyond. https://proceedings.mlsys.org/paper/2019/file/1d7f7abc18fcb43975065399b0d1e48e-Supplemental.pdf

Wells, L., Bednarz, T. Explainable AI and reinforcement learning—a systematic review of current approaches and trends. (2021). https://doi.org/10.3389/frai.2021.550030