Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
COVID vision: An integrated face mask detector and social distancing tracker

Janvi Prasad, Arushi Jain, David Velho, Sendhil Kumar K S

School of Computer Science and Engineering, VIT University, Vellore, India

1. Introduction

COVID-19 has immensely affected the world. Growth in fields like Agriculture, Industry and Finance have halted or dropped significantly and can only resume with strict rules and regulations, such as wearing masks and maintaining a minimum distance of 3 feet between two people. Enforcing these rules and regulations is mandatory and essential for safety but is a difficult task since it requires a large number of dedicated personnel as well as public cooperation. As this is not always feasible, we have identified the need for an automated system to detect certain COVID-19 violations in real time. Our system can scan people’s faces to detect the presence of a mask and if it is being worn to an acceptable standard, determine if the distance between two people is below the minimum required distance and also maintain a database of those who have tested positive for COVID-19 or are at risk using facial recognition.

Computer-Vision based automatic detection and control systems are easy, effective and economic solutions to check the spread of COVID-19 in public areas. Although the idea is straightforward, the design and implementation of such systems require extensive system design since this system must be fast and real-time to detect social distancing infractions and immediately send a warning. Socio-ethical considerations such as privacy concerns also need to be made. These issues can be alleviated with a real-time system which does not store sensitive image data and only keeps aggregate statistics, such as the number of daily violations.

With such a surveillance system, appropriate measures can be taken as quickly as possible to reduce further spread of COVID-19.

Computer Vision is a derivation of Artificial Intelligence where computers extract details and other data of interest from an image or video using specialized algorithms. Thus, computer vision can be extended to many applications, with varying degrees of accuracy, depending upon the use cases. Our project on Face Mask Detection & Social Distancing Tracker uses computer vision to understand various aspects of the images or videos based on frames that would be provided as an input to the algorithms. The primary concept that is used behind this is to find the bounding boxes related to the classes depending on the training data sets.

CNNs have become the de facto norm in a variety of computer vision tasks due to their inherent ability to work well with spatial data. CNNs work on the concept of developing an internal representation of 2 important aspect of 2D images - the position and scale. Therefore, we use a sequential CNN as the foundation for our face mask detector along with core ML packages such as TensorFlow, Keras and OpenCV.

YOLO is a popular framework for live object detection and is used to find the bounding boxes for our social distancing tracker. Its high speed, accuracy and learning capabilities makes it a good choice for the real-time nature of our project. LBPH (Local Binary Pattern Histogram) is one of the best algorithms to represent local features in images and is especially suited for facial recognition. Haar cascade classifiers identify faces in an image or a real time video. They use the concept of integral images, where every pixel is the summation of the pixels above and to
the left of it, and the AdaBoost algorithm (also known as the Adaptive Boost Algorithm) which make them very efficient. These two algorithms are used in conjunction for our face recognition module.

2. Related work

During face detection, a face is detected from an image that includes several other attributes of interest in it. Given an input image, the challenge is to identify and extract the face from the picture. Face detection is a difficult task because the faces differ in size, shape, color, etc. and they are not immutable.

Obstructive face detection comes with two major challenges: (1) unavailability of large datasets containing both masked and unmasked faces, and (2) exclusion of facial ex-pression in the area covered by a mask. Utilizing the locally linear embedding (LLE) algorithm and dictionaries trained on an immensely large collection of masked faces, several lost expressions can be recovered and the reliance on facial cues can be reduced to a great extent. According to the work reported by Yamashita et al. (2018), convolutional neural networks (CNNs) in computer vision come with a strict constraint regarding the size of the input image. Common practice consists of reconfiguration of the images before fitting them into the network. The main goal is to detect a person’s face from the image correctly and then identify if it has a mask on it or not. In order to perform the task in real-time, the proposed method should also detect a face along with a mask in motion, accounting for motion blur. In the last decade, convolutional neural networks (CNN), region-based CNN and faster region-based CNN used region proposal techniques to generate the objectness score (how likely it is to be an object of interest) prior to its classification and later generates the bounding boxes around the object of interest for visualization and other statistical analysis as proposed by Reshma Prakash & Nath Singh (2022) & Ren et al. (2015).

Although these methods are efficient, they suffer in terms of larger training time requirements. Since all these CNN based approaches utilize classification, another approach, YOLO, considers a regression-based method to dimension-ally separate the bounding boxes and interpret their class probabilities. In this method, the designed framework efficiently divides the image into several portions representing bounding boxes along with the class probability scores for each portion to be considered as an object. This approach offers excellent improvements in terms of speed. The detector module exhibits powerful generalization capabilities of representing an entire image. Chen et al. (2021) & Garg et al. (2018), Yang & Jiachun (2018) use YOLO for face detection in their work.

Nawaf Hazim et al. (2016) proposed an automatic face recognition system developed using exterior-based methods. The Viola-Jones method is used to detect and gather faces in each database. Square Euclidean Distance is used to calculate the distance between two images, which leads to the image similarity. Ningtoujum Sunita Devi (2014) presented a methodology for face recognition based on the information theory approach of coding and decoding the face image. Proposed methodology consists of two stages – Feature extraction using principal component analysis and recognition using the feed forward back propagation Neural Network.

The proposed method has been tested on an Oracle Research Laboratory (ORL) face database containing 400 images. Jignesh Chowdary et al. (2020) built a face mask detector by using transfer learning of the pretrained InceptionV3 model. The system uses the Simulated Masked Face Dataset (SMFD) and image augmentation techniques. Qin & Li (2020) divided the face mask conditions into three classes: correct covering, incorrect covering, and no face mask. Their proposed system uses SRCNet to implement image super-resolution and classify images. Rusli et al. (2020) proposed an android app for tracking social distancing. It uses GPS and Bluetooth Low Energy (BLE) and sends alerts to the user’s phone. Drawbacks in the system stem from the variations in distance estimation using BLE values, subject to various factors. Kumar et al. (2019) and Hjelmås (2001) discuss a comprehensive survey of various techniques used for face detection and the different challenges and applications for each technique are also given. Furthermore, it lists some standard databases for face detection along with their features. Yang et al. (2016) propose the WIDER FACE dataset to help with face detection research in the future. It is 10 times larger than existing datasets and richly annotated. The authors also benchmark several representative existing detection systems.

Zhang et al. (2021) proposed stochastic pooling to replace average pooling and max pooling, combined conv layer with batch normalization layer and obtained the conv block and combined dropout layer with fully connected layer and obtained the fully connected block. Abbasi et al. 2021 introduced a new dataset, and two distinct methods for detecting masked and unmasked faces in real-time. An object detection model is used in the first approach. In the second method, a YOLO face detector detects faces (whether masked or not), and then uses a novel fast yet effective CNN architecture to classify them into the two categories. Loey et al. (2021) proposed a model that is made up of two parts. The first part is based on the ResNet-50 deep transfer learning model and is geared for feature extraction. The second part is based on YOLO v2 and is meant to detect medical face masks.

To offer an upgrade for face detection systems, Aung et al. (2021) have integrated the YOLO technique with the VGG16 pretrained convolutional neural network and Girshick (2015) has demonstrated how quickly R-CNN can train end-to-end detectors on shared conv features, with impressive accuracy and speed. Researchers, Dai et al. (2015), He et al. (2015), & Sermanet et al. (2013), have shown that, for efficient yet accurate visual identification, shared computation of convolutions is gaining popularity.

To diagnose COVID-19, many researchers, Magshed et al. (2020), Wang et al. (2020a) & Manapure et al. (2020) have used deep learning methods. Magshed et al. (2020) demonstrated an affordable method of diagnosing COVID-19 using smartphone-embedded sensors. Wang et al. (2020a) proposed a deep learning system for diagnosing COVID-19 patients using chest radiography scans.

Manapure et al. (2020) have presented a deep learning system for automatic segmentation and quantification of infection regions as well as the entire lung from chest CT scans. Wang et al. (2020b) proposed a novel method for the automatic detection of COVID-19 from chest CT images using wavelet Renyi entropy, feedforward neural network and a proposed three-segment biogeography-based optimization (3SBBO) algorithm.

3. Technology used

3.1. TensorFlow

TensorFlow is an interface that is used for expressing machine learning algorithms and implementing ML systems across a variety of computer science fields, that include: sentiment analysis, geographic information extraction, computational drug discovery, computer vision, voice recognition, information retrieval and text summarization. TensorFlow is used at the backend of our COVID Vision model’s sequential CNN architecture (consisting of numerous layers). It is also used in data processing to restructure the data or image.

3.2. Keras

Keras provides essential building units for the design and transfer of machine learning systems at high iteration speeds. Tensorflow’s scalability and cross-platform features are fully utilized. Das et al. (2020) tell us that Keras’ primary data structures are layers and models. Keras is used to implement all the layers in the CNN model. With the conversion
of the class vector to the binary class matrix in data processing, it aids in the compilation of the overall model.

3.3. OpenCV

OpenCV (Open Source Computer Vision Library), an open-source computer vision and Machine Learning frame-work, is utilized to differentiate and recognize faces, recognize objects, group movements in recordings, trace progressive modules, follow eye movements, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database using similarity algorithms, perceive landscapes and set up markers to overlay them with increased reality and so forth.

COVID Vision makes use of these features of OpenCV in preprocessing and conversion of image data.

3.4. Regression-based object detectors (YOLO)

Regression-based object detectors like — You Only Look Once (YOLO) and Single Shot Detector (SSD) multi-box have been proven to be significantly faster than region-based object detectors. Among the two and other similar object detectors, YOLO has long been the most popular choice. It takes as input the entire image at once, unlike region-based detectors which deduce region proposals that are fed to the classifier. The model pipeline expects an RGB image which is divided into grid cells $S \times S$. Each grid cell is responsible for predicting B bounding boxes. For each bounding box, 5 values are predicted x, y, w, h, and c. The coordinates of the center point of a bounding box relative to a grid cell are x and y, and the width and height of the bounding box are w and h. The confidence score of an object being present in a bounding box is c. For class probabilities, c, the output of the object detector is a tensor of shape.

4. COVID vision architecture

COVID Vision consists of the following components:

4.1. Face mask detector

The main challenge of any face mask detection task is twofold - we first need to detect the face from the image correctly and then identify if it has a mask on it or not. The problem is similar to general object detection, which is used to identify different types of objects. Face identification is the process of categorizing and differentiating a certain group of items, namely faces. It has a variety of applications, including autonomous driving, surveillance, etc. The core Machine Learning (ML) packages such as TensorFlow, Keras, and OpenCV are used in this research to propose a simplified solution to suit the aforesaid objective. In order to perform surveillance tasks, the proposed method should also detect a face along with a mask in motion.

4.2. Social distancing tracker

YOLO stands for You Only Look Once and is used for Object Detection as well as Object Tracking. In our proposed solution, we use YOLO for calculating the social distance with the help of Object Detection, and tracking the people in the frame, counting the objects, and keeping a record of that object in the next frame by Object Tracking. The minimum distance to keep while adhering to social distancing is 6 feet, and we keep this as the base for calculating distance while training the model.

We built our framework on the YOLO v3 model, and the model was trained on the COCO dataset which has a total of 64,115 images of people. Every bounding box object that is detected is the architecture output along with its score of confidence. YOLO could make the prediction of the class and position of an object just by seeing the image one time.

YOLO takes into account object identification issues as regression jobs in place of classification to allocate anchor class probability boxes.

Other comparable object detection algorithms like SSD (Single Shot MultiBox Detector) and FRCNN (Faster R-CNN) exist. On comparison of these algorithms using the COCO dataset in identical testing environments on parameters such as accuracy, precision and F1 score by Srivastava et al. (2021), it was found that YOLO-v3 outperforms SSD and Faster R-CNN. On comparing the algorithms on the basis of a pill recognition model for detecting difficult samples after training each of them on a pill image dataset by Tan et al. 2021, it is found that the mean average precision (MAP) of Faster R-CNN reached 87.69% but YOLO v3 had a significant advantage in detection speed where the frames per second (FPS) was more than eight times than that of Faster R-CNN. This means that YOLO v3 can operate in real

4.3. Face recognition

In this system, face recognition for human faces is implemented using LBPH for feature extraction and Haar cascade classifier for face detection. In general, the procedure for face recognition can be divided into the following tasks: establishing a dataset, face acquisition, feature extraction, and classification. Our entire project is written in OpenCV with python.

On comparing the different available face recognition algorithms like Eigenface which uses Principal Component Analysis (PCA), Fisherface which uses Linear Discriminant Analysis (LDA) and LBPH which uses histograms, by SudhaNarang & MeghaSaxena, 2018, with a view of Attendance Management, they found that LBPH outperforms the other algorithms with confidence factor in range 2--5 and has minimum noise interference and maximum efficiency. The Eigenface and Fisherface algorithms use component-based principal of dataset generation while the LBPH uses pixel-based principal of dataset generation.

5. COVID vision methodology

5.1. Face mask detector

In a variety of computer vision tasks, CNNs have become the de facto norm. Sequential CNN is used in the current technique. The Rectified Linear Unit (ReLU) and MaxPooling levels come after the first convolution layer. A total of 100 filters are used in the convolution layer. The height and width of the 2D convolution window are specified by the kernel size, which is set to $3 \times 3$. The Conv2D class’s activation parameter is set to "relu." It depicts a nearly linear function with all of the advantages of linear models and the ability to be easily optimized using gradient-descent methods. When compared to other activation functions, it performs better in terms of performance and generalization in deep learning.

To lower the output volume’s spatial dimensions, Max Pooling is used. MaxPooling is preferred over stochastic pooling since you may lose main features of the inputs because in comparison with max pooling, here, you cannot predict which parts of the input will be chosen. The second and the third convolutional layers are defined in a similar way to the first layer. The long vector of input is routed via a flatten layer, which converts a matrix of characteristics into a vector that can be inputted into a fully connected neural network classifier, to feed the data into the CNN. The Softmax activation function is used in the final layer (Dense), which has two outputs for the two categories - With Mask and Without Mask.

5.2. Social distancing tracker

The YOLO v3 model is trained using the COCO dataset. In the first stage, we apply our detection model to this dataset to extract bounding boxes for each person in a frame. The next step is detection and classifying the output of YOLO v3. Normally it operates on bound boxes, but
we illustrate only the center of the bounding box for simple visualization. After this stage, we measure the Euclidean distance. To measure this distance, we need to extract the point coordinates of each center. This operation needs to be done for all frames in the sequence. Then, we take the absolute value of the output number. The main outputs from these frames are their x and y coordinates. In the final stage, we measure the Euclidean distance using x and y coordinates. Euclidean distance is formulated by:

\[ d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \]

Using this formulation, we can measure the distance between detected humans in the frame sequences. After measuring the Euclidean distance, we should keep a minimum threshold for human-to-human distance:

5.3. Face recognition

We train the captured images using LBPH face recognition technique. LBPH or Local Binary Pattern Histogram is one of the easiest face recognition algorithms. It can represent local features in the images and gets great results (mainly in a controlled environment). It is robust against monotonic gray scale transformations. It is a built-in function provided by the OpenCV library. The LBPH uses 4 parameters:

1. Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel.
2. Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind - the more sample points you include, the higher the computational cost.
Grid X: the number of cells in the horizontal direction. The more the cells, the finer the grid, the higher the dimensionality of the resulting feature vector.

Grid Y: the number of cells in the vertical direction. Same principle for increasing the number of cells as Grid X.

Face recognition system encompasses three main phases which are face detection, feature extraction, and face recognition.

1. Face Detection: Face acquisition and localization from an image is detected using a Haar cascade classifier.

Preprocessing of human faces are separated from the objects present in an image.

1. Feature Extraction: From the detected face we are extracting the features through LBPH. First we compute the local binary pattern images and then histograms are created.

2. Face Recognition: The extracted features are fed to the LBPH model we trained using the photos we captured earlier and this model thus recognizes or classifies by using machine learning algorithms.

6. Results and analysis

6.1. Face mask detector

Hence with the help of TensorFlow and CNN models embedded with image processing, we were able to train our model to accurately detect a mask on a face.

The method attains an accuracy of upto 95.77%. We achieve this accuracy reading due to MaxPooling. It provides rudimentary translation invariance to the internal representation along with the reduction in the number of parameters the model has to learn. This sample-based
discretization process down-samples the input representation consisting of the image, by reducing its dimensionality.

Number of neurons have the optimized value of 100 which is not too high. A higher number of neurons and filters can lead to worse performance. The optimized filter values and pool size help to filter out the main portion (face) of the image to detect the existence of the mask correctly without causing over-fitting. The face mask detector module was tested on 150 people.

Our model can process a frame in approximately 55 ms, giving us an average frame rate of 18.1 fps. In the real-time experimentation, the model shows 100% accuracy. If a person covers their mouth with their hands, it will show that no mask is detected.

6.2. Social distancing tracker

The results provided are acquired from COVID Vision during its runtime. All the results are based on a model trained on COCO dataset. We illustrated the detected pedestrians and alert levels with green and red bounding boxes. The model will make a line between 2 humans, even if the distance between them is a little bit lower than the threshold. This regulates in which specific area there is a social distancing violation, through the live feed. The threshold is fixed to 50 based on our dataset angle which was projected during the video capturing. There is no fixed threshold, and it can be changed based on the camera angle.

If the camera is faced at larger distances, the threshold should be considerably reduced because the accuracy declines as the camera is placed far away from the crowd.

To find the accuracy of our social distancing tracker, live footage was taken from 3 CCTV cameras at separate settings. The first CCTV camera was set up in a college, the second one was in a cafe, and the third one in a public area. Footage from each of these three cameras was tested for 2 h. The footage was also human annotated by three different reviewers to find the accuracy of the social distancing tracker. The mean of the observations by three different reviewers was calculated. The average social distancing violations were found to be 86.67, 54.33 and 125.33 for the three different settings respectively. Comparing these values to our social distancing tracker, we find the accuracy of our system to be 96.49%.
6.3. Face recognition

The results provided are acquired from COVID Vision during its runtime. We get the results after training the model with the captured images using the LBPH face recognition technique. Once the training is complete, we generate a model to recognize faces. Finally, add the details to the database.

While testing the face mask detector, we also captured images of 150 people to train our face recognition model. After training the model using the LBPH face recognition technique, we get the results. The model was able to identify 135 faces correctly. Thus, the model is 90% accurate.

7. Conclusion

To lessen the spread of the COVID-19 pandemic, several measures should be taken. We have demonstrated a face mask detector using the sequential model of the convolutional neural networks (CNN). We later demonstrated a social distancing tracker using the DNN model as well as the COVID patient’s face recognition using LBPF face recognition technique.

The outputs were promising since our system achieved high levels of accuracy, and hence we can say that the code is effective and can be deployed for real-time monitoring. This system can be implemented in numerous locations like shopping centers, airports, and other high-traffic places to screen people and to reduce the spread of the infection by checking who is following essential rules and who is not. It can also be connected to an alarm or a buzzer that would notify owners or other administrators that there are users without masks and can hence ensure the safety and well-being of other people in the surrounding area.

Figs. 1–13.

Future work

Further we will work to classify the faces into more categories like “Improper Mask” instead of just the two main classes - with and without mask. We can achieve this by adding datasets with images of people wearing masks that are not covering their noses properly. We can also detect masked faces using the FaceNet model of Convolutional Neural Networks so as to further improve our model.

Although YOLO v3 outperformed the other state-of-the-art methods, there are several limitations to this model. The video quality is an essential factor for providing an accurate assessment and the depth factor is important too. We aim to validate the calculated distances at each frame. One natural application of our framework could be flagging the violations of social distancing with a fixed threshold and creating an alarm mechanism.

Additionally, for basic surveillance, we have decided to implement this system in an android application which can further be extended to surveillance cameras to keep track of non-observance of COVID-19 restriction in public and open spaces where there is more crowd.

The future work lies in deploying mask detection and temperature detection of each person in a crowd. Combining multiple models in an ensemble learning framework can be instrumental in an end-to-end pipeline for complete monitoring of each person.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abbasi, S., Abdi, H., & Ahmadi, A. (2021). A face-mask detection approach based on YOLO applied for a new collected dataset. In Proceedings of the 26th international computer conference, computer society of Iran (CSCI) (pp. 1–6). IEEE.

Aung, H., Bohlov, A. V., & Tun, N. L. (2021). Face detection in real time live video using YOLO algorithm based on Vgg16 convolutional neural network. In Proceedings of the international conference on industrial engineering, applications and manufacturing (ICIEAM) (pp. 697–702). IEEE.

Chen, W., Huang, H., Peng, S., et al., (2021). YOLO-face–A real-time face detector. The Visual computer, 37, 805–813. 10.1007/s00371-020-01831-7.

Dai, J., He, K., & Sun, J. (2015). Convolutional feature masking for joint object and stuff segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3992–4000).

Das, Atya, Anari, Mohammad Wanif, & Bank, Rohini (2020). COVID-19 face mask detection using TensorFlow. Keras and OpenCV. 10.13140/R.G.2.2.21473.94565.

Garg, D., Goel, P., Pandya, S., Ganatra, A., & Kotecha, K. (2018). A deep learning approach for face detection using YOLO. In Proceedings of the IEEE punecon (pp. 1–4). IEEE. 10.1109/PUNECON.2018.8745376.

Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440–1448).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904–1916.

Hjelmé, E. (2001). Face detection–A survey. Computer Vision and Image Understanding, 83(3), 236–274 ISSN 1077-3142. 10.1006/cviu.2001.0921.

Jignesh Chowdhary, G., et al., (2020). Face mask detection using transfer learning of inceptionv3. In Proceedings of the international conference on big data analytic. Springer Cham.

Kumar, A., Kaur, A., & Kumar, M. (2019). Face detection techniques–A review. Artificial Intelligence Review, 52, 927–949. 10.1007/s10462-018-9650-2.

Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2021). Fighting against COVID-19–A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. Sustainable Cities and Society, 65, Article 102660.

Maghded, H. S., Ghafoor, K. Z., Sadig, A. S., Curran, K., Rawat, D. B., & Babie, K. (2020). A novel AI-enabled framework to diagnose coronavirus COVID-19 using smartphone embedded sensors–Design study. In Proceedings of the IEEE 21st international conference on information reus and integration for data science (IRID) (pp. 160–167). IEEE.

Manapure, P., Likhar, K., & Kousare, H. (2020). Detecting COVID-19 in X-ray images with keras, tensor flow, and deep learning. Assessment, 2(3).

Nawfal Hazim, B., Sinan Sameer, M. A., Wael Esam, M., & Mustafa Abdul Sahib, N. (2016). Face detection and recognition using Viola-Jones with PCA-LDA and square Euclidean distance]. International Journal of Advanced Computer Science and Applications, 7(5).

Ninghousam Sunita Devi, K. (2014). Hemachandran, –Face recognition using principal component analysis] 5: (pp. 6491–6496).
Qin, B., & Li, D. (2020). Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. Unpublished results, doi: 10.21203/rs.3.r2b8668/v1

Rahma Prakash, S., & Nath Singh, P. (2022). Object detection through region proposal based techniques. Materials Today: Proceedings, 10.1016/j.matpr.2021.02.533

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39. 10.1109/TPAMI.2016.2577031.

Rusli, M. E., Yussof, S., Ali, M., & Abobakar Hassan, A. A. (2020). MySD-A smart social distancing monitoring system. In 2020 8th international conference on information technology and multimedia (ICIMU) (pp. 399–403). 10.1109/ICIMU49871.2020.9243569.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.

Srivastava, S., Divyakar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Patabhiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. Journal of Big Data, 8(1). 10.1186/s40453-021-00434-w.

Ts, I. n, Huangfu, Tianran, & Wu, Liyao et al. (2021). Comparison of YOLO v3, Faster R-CNN, and SSD for Real-Time Pill Identification, 30 Jul, PREPRINT (Version 1) available at Research Square [10.21203/rs.3.rs-668895/v1]

SudhaNarang, K. J., & MeghaSaxena, A. A. (2018). Comparison of face recognition algorithms using OpenCV for attendance system. The International Journal of Scientific and Research Publication, 8(2) (ISSN: 2250-3153). http://www.ijsrc.org/research-paper-0218.php?rp=P747200.

Wang, L., Lin, Z. Q., & Wong, A. (2020a). COVID-net–A tailored deep convolutional neural network design for detection of COVID-19 cases from chest x-ray images. Scientific Reports, 10(1), 1–12.

Wang, S. H., Wu, X., Yu-Dong, Z., Tang, C., & Zhang, X. (2020b). Diagnosis of COVID-19 by wavelet Renyi entropy and three-segment biogeography-based optimization. International Journal of Computational Intelligence Systems, 13(1), 1332.

Yamaishi, R., Nishio, M., Do, R.K.G., & Togashi, K. (2018b). Convolutional neural networks–An overview and application in radiology. Insights into Imaging

Yan, S., & Jiachun, Z. (2018). Real-time face detection based on YOLO. Proceedings of the 1st IEEE international conference on knowledge innovation and invention (ICKII) (pp. 221–224). IEEE.

Zhang, Y., Satapathy, S. C., Liu, S., & Li, G. R. (2021). A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. Machine Vision and Applications, 32(1), 1–13.

Jani Prasad is a student pursuing her B.Tech. in Computer Science and Engineering at VIT, Vellore. She has interests in applications of emerging technologies like AI/ML and IoT and has presented/published papers in reputed conferences. She is a member of the Google Developers Student Club in VIT.

Arushi Jain is a third-year undergraduate, currently pursuing her B.Tech. in Computer Science and Engineering at VIT, Vellore. She is an experienced IoT and web developer and has built many projects to combat prevalent social issues. She has been a part of the Google Developers Student Club and the Computer Society of India in VIT. Her research interests include Machine Learning, Internet of Things, Computer Vision and Cyber Security.

David Velho is a fourth-year undergraduate studying Computer Science and Engineering from VIT, Vellore. He is currently working as an intern at the Institutional Shareholder Services - Governance. He is an experienced web, computer vision and systems developer and has contributed to multiple open-source projects. He has been the Technical Head of Computer Society of India at VIT. His research interests include Machine Learning and Cyber Security.

K.S Sendhil Kumar works as an Associate professor in School of Computer Science and Engineering at VIT, Vellore. He received his Ph.D. in Computer science and Engineering from VIT, Vellore, ME (CSE) from Anna University and BE (CSE) from University of Madras. He has 15 years of experience in teaching and research. He has published many papers in international and National Journals and conferences on Cloud computing, Machine Learning, IoT and Big Data. He acts as a reviewer for many international journals and conferences. He is a life member of Computer Society of India. His research interests include cloud computing, Block Chain, IoT and big data.