# Facial Mask Detection and Face Identification using CNN with Transfer Learning

**Chandrika Saha**[1], **Noshin Tasnim**[2], **Sareh Soltani Nejad**[3]

{csaha[1], ntasnim3[2], ssolta7[3]}@uwo.ca

Department of Computer Science

The University of Western Ontario

December, 2021

# Contents

# 1 Introduction

The year 2019 has brought a tremendous change in our lives with the first report of the infectious coronavirus disease (COVID-19). The increase in the number of COVID-19 positive tests all over the world has led to confinement and lockdowns, lockdowns by the governments of most of the countries of the world. It has been discovered that some non-pharmaceutical precautions can help to lower down the chance of the COVID-19 virus spreading. Wearing a facial mask is one of the precautions as covering nose and mouth with facial mask can reduce Coronavirus spread by avoiding forward distance travelled by a person's exhaled breath by more than 90% [8].

After the increasing rate of COVID-19 vaccine intake, the world starts going through various stages of reopening with maintaining strict safety measurements. One of the safety measures is to wear facial masks while they are in indoor public places. Although, a number of people still don't abide by this rule. Some organizations and places have appointed staff to monitor if people are following the rule of wearing facial masks.

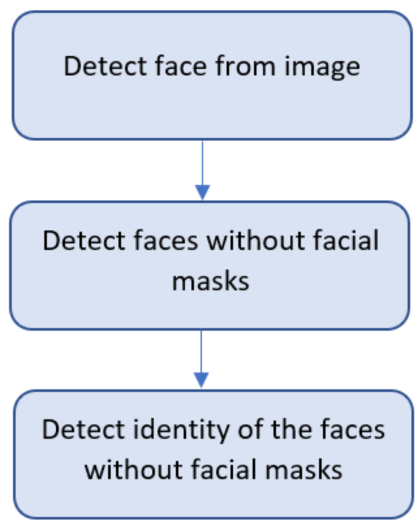


Figure 1: Sysyem work flow

In this era of technological development, it is high time to make this facial monitoring system more developed and digitalized. Therefore, our work aims to build a facial mask detection system that can be implemented to detect people who are not wearing masks in public places such as airports, university buildings, shopping malls, superstores etc. Along with that, implementation of a facial identification detection system along with the facial mask detection system will help to keep track of the identity of those people who do not wear facial masks in order to alert them about wearing facial masks in public places.

To develop a working facial mask detection and identity detection system, we have to proceed through three stages of detection systems. First, we have to detect face from image. Second step is to detect the presence of a mask on the detected face. After that, if a face without a facial mask is detected, the system will detect the identity of that person.

There are several machine learning algorithms which has been used for face detection previously for security or biometric purpose. In this paper, we will develop the whole detection system by implementing Convolution neural networks (CNN) model as it provides excellent performance for computer vision and image processing. With a view to improve the performance of the model, we will also use transfer learning, which has been proved to help improving the performance of machine learning models by using pre-trained classifiers.

## 2 Literature Survey

Since COVID, mandatory facial mask wearing has been a recent issue. Therefore, there has been some recent work regarding this. In 2021, Shamrat *et al.* [16] have proposed a model for facial mask detection using three deep learning methods including Max pooling, Average pooling, and MobileNetV2 architecture. They have used a dataset containing 1845 images from various sources with a webcam and a mobile phone camera. The Max pooling achieved 98.67% accuracy, the Average pooling achieved 96.23% accuracy and MobileNetV2 architecture gained 99.82% accuracy. In 2020, A. Das *et al.* [4] proposed a simplified approach to detect facial mask using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn and achieved accuracy up to 95.77% and 94.58% respectively on two different datasets.

Another work has been done by S. Sethi *et al.* [15] in 2021 where they have used three models of transfer learning - ResNet50, AlexNet and MobileNet and achieved highest accuracy (98.2%) when implemented with ResNet50. In 2021, P. Nagrath *et al.* [13] have proposed a model SSDMNV2 for face mask detection using OpenCV Deep Neural Network (DNN), TensorFlow, Keras, and MobileNetV2 architecture. They have used OpenCV Deep neural networks in SSDMNV2 which have SSD with ResNet-10 as backbone and is capable of detecting faces in most orientations. And for lightweight and accurate predictions, they have used MobileNetV2 architecture.

Boulos, Mira M. [2] in 2021, presented a work on facial recognition and face mask detection system through deep learning techniques by building a CNN model using transfer learning and fine-tuning techniques and achieve a 97.1% of performance accuracy.

Face Identification has been a very popular topic for a while. There has been some great work on Face identification for years. In 2021, R. J.Hassan and A. Abdulazeez worked on a review for a face recognition model using deep convolutional networks [9]. In that paper, they have showed the flow of recent published work where use of different methods and architectures such as (SGD, LBP, SVM, PCA, Haar-cascade, Dropout) and (AlexNet, MobileNet, VGG16, SqueezeNet, ShuffleNet, LeNet, IMISCNN) has been shown. In 2021, H.Ge *et al.* have used multi- task convolutional neural network for face recognition [7]. In their model, they trained it on a database of 50 faces that they collected and used PSNR, SSIM and ROC curve to analyze MTCNN, R-CNN and Faster R-CNN. In another work in 2021, KH Teoh *et al.* [17] came up with a concept of a face recognition system where they worked on a deep learning approach using OpenCV in python. Mishra *et al.* [12] also worked on a 3D CNN for face recognition with DenseNets performing the best with an accuracy of 97% on CVBL dataset.

Looking at the works of 2020, Wang *et al.* [19] proposes a face recognition algorithm based on improved convolutional neural network. In that year another work has been done by L. Wang *et al.* [20] where convolution neural network (CNN) is used to solve the problem of low efficiency of deep learning in big data processing. Based on this point, a face recognition system based on CNN was designed. M. Farayola *et al.* proposed CNN based model on a pre-trained VGG Face for face recognition from a set of faces tracked in video or image capture achieving a 97% accuracy in 2020 [5]. Yao [21] proposed a compact convolutional neural network for face recognition. Their work not only provides a compact foundation for DL, but it also significantly improves the efficiency of face recognition. During the implementation process, they use the Caffe toolkit for preparation and rollout. They achieved an incredibly successful method for the traditional face verification pipeline, with 94.8 percent face verification precision on LFW. The comparison results on different face recognition tasks suggest that the proposed compact CNN structure could be used in face recognition systems.

Another promising work in 2019 was done by R.M. Prakash *et al.* [14] where they worked on an automated face recognition method using Convolutional Neural Network (CNN) with transfer learning approach. They pre-trained model VGG-16 on a huge ImageNet database and used the knowledge in their CNN model. They extracted features were fed as input to the Fully connected layer and softmax activation for classification.

The Siamese neural network has become a popular method in face recognition. In [11], face recognition was achieved by training a convolutional network (CNN) with aligned 3D facial data, allowing the CNN to detect faces regardless of facial expressions. After that, a Siamese network shared the weighted trained CNN in order to create a 3D facial recognition model that can recognise faces even with a tiny sample size. Their proposed method achieved a recognition rate of 0.977 on the FRGC database, indicating that it can be used for facial recognition.

Jianming Zhang *et al.* [23] proposed a strategy for achieving 94.8 percent accuracy in face recognition by training their model with the small-samples dataset LFW by constructing a new CNN and applying it in siamese architecture. The proposed method in [10] is similar to Jianming Zhang *et al.* [23] with siamese network architecture. Using transfer learning and the VGG-16 Model, which is pre-trained on the ImageNet dataset, they enhance accuracy by up to 95.2 percent. Face recognition is implemented in this paper using a siamese network architecture, which consists of two similar CNN networks- and transfer learning. A pair of face images is given into the proposed model, and the network evaluates whether the images belong to the same person or not by extracting features from the images and producing a similarity index.

# 3 Methodology

## 3.1 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNNs) are a type of neural network that has shown to be highly successful in image recognition and classification. They are commonly used to analyze visual imagery and are frequently working behind the scenes in image classification. CNNs are multi-layered feed-forward neural networks. They are made up of filters, kernels, or neurons with weights, parameters, and biases that may be learned [18]. Each filter takes input images and performs convolution to extract features [22]. A typical CNN architecture is shown in Figure 1 [3].

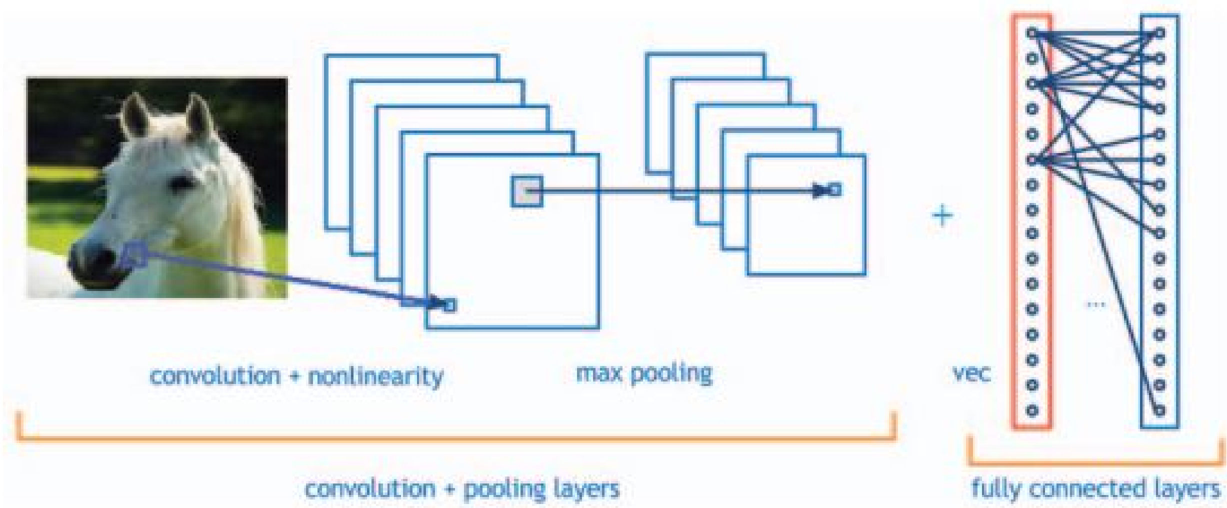


Figure 2: A Convolutional Neural Network architecture

1. **Cnvoutial Layer**
   The main component of a Convolutional Network is the Convolutional Layer. It is employed in a method known as Feature Extraction to extract various characteristics from the input images. Between the input image and a filter, this layer performs the mathematical action of convolution. The pixels in a convolution's receptive area are all converted into a single value. The dot product between the filter and the sections of the input image with regard to the size of the filter is calculated by sliding the filter across the input image. The convolutional layer's final output is a Feature map vector, which contains information about the image's corners and edges. This feature map is then supplied to further layers, which learn a number of different features from the input image.

2. **Pooling Layer**
   This layer's major goal is to lower the dimensionality of the convolved feature map in order to reduce computational costs. This is accomplished by reducing the connections between layers and operating on each feature map individually. Max pooling is performed

by the pooling layer, which takes the highest value in a given filter zone. The Pooling Layer is typically used to connect the Convolutional Layer and the FC Layer.

3. **ReLU Layer** ReLU ((rectified linear unit) is a non-linear activation function for many types of neural networks. This layer is used to increase non-linearity in the network. It removes negative values from an feature map by setting them to zero.

4. **Fully Connected Layer**
   The Fully Connected (FC) layer, which includes the weights and biases as well as the neurons, is used to link the neurons between multiple layers. The FC layer receives the flattened input picture from the previous levels. Following that, the Flattened vector is connected to a few fully connected layers, which conduct the same mathematical operations. The classification procedure starts at this point. The final layer employs the softmax activation function to derive probability of the input being in a given class after passing through the fully connected network.

5. **Softmax**
   In CNN's output layer, which predicts a multinomial probability distribution, the softmax function is used. Softmax is the activation function for multi-class classification problems. Although less popular, the function can be utilised as a hidden layer activation function in a neural network.

### 3.2 Transfer Learning

In real world, most of the time machine learning methods suffers for lack of data. It is relatively hard and expensive to have a dataset of sufficient size. Moreover, to build a powerful machine learning model, we require high computational resources. Even with high computational resources, it also takes a lot of time for this task. To accumulate these issues, Transfer learning has been introduced. Transfer learning is a concept where a learning algorithm reuses the knowledge from the past related tasks to ease the process of learning to perform a new task. The main concept is to take a source domain with large dataset and computational resources and train the model. After training, the gathered knowledge from the source domain can be transferred to a target domain with a smaller dataset and limited computational resources.

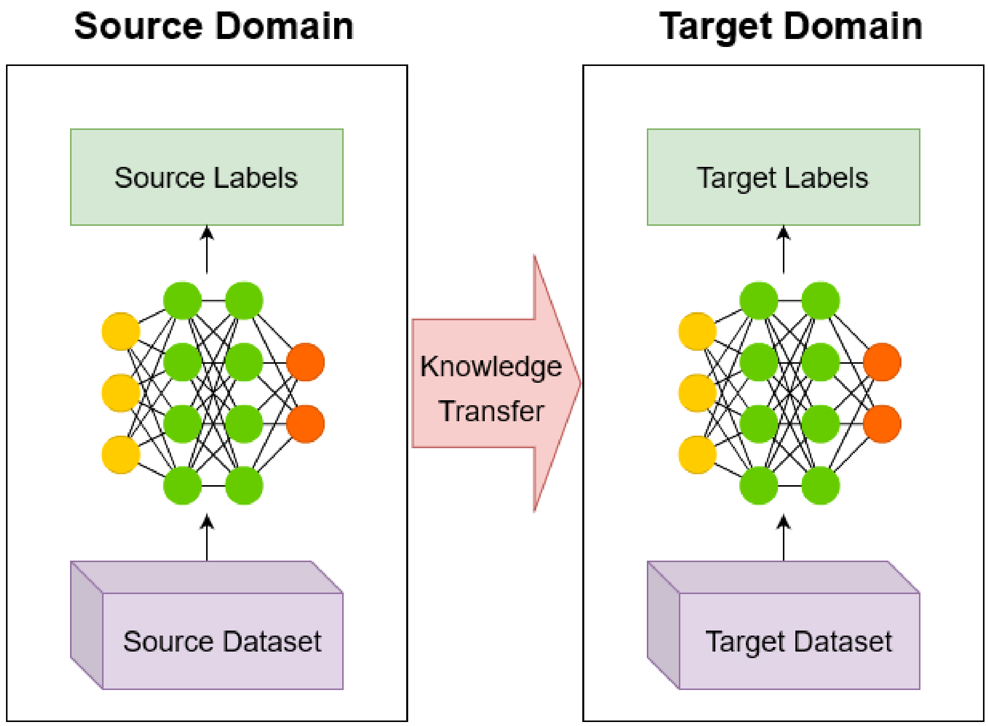


Figure 3: Transfer Learning Approach

For Convolutional network, Transfer learning works by using the previous neural network model to identify edges in the earlier layers, structures in the middle layer, and high-level features in the later layers [2]. Therefore, Using transfer learning, it is not necessary to train an entire Convolutional Network from scratch, we can use the knowledge of a pre-trained model to train a new model faster.

### 3.3 Siamese Network

Nowadays many machine learning algorithms are being used for study related to images and achieving high performances as well. However, most of the learning algorithms require large amount of data. A learning algorithm doesn't train well with less amount of data. In this scenario, Siamese networks performs very well. It can give a good accuracy rate even if it is trained with less amount of data from imbalance distribution.

The Siamese network consists of a pair of Neural Networks which are identical to each other, also known as Sister Networks. Unlike a traditional CNN, the Siamese Network does not classify or label images; instead, it simply calculates the distance between any two images. If the images have the same label, the network should learn the parameters, such as weights and biases, so that the distance between them is lower; if the images have different labels, the distance should be bigger.

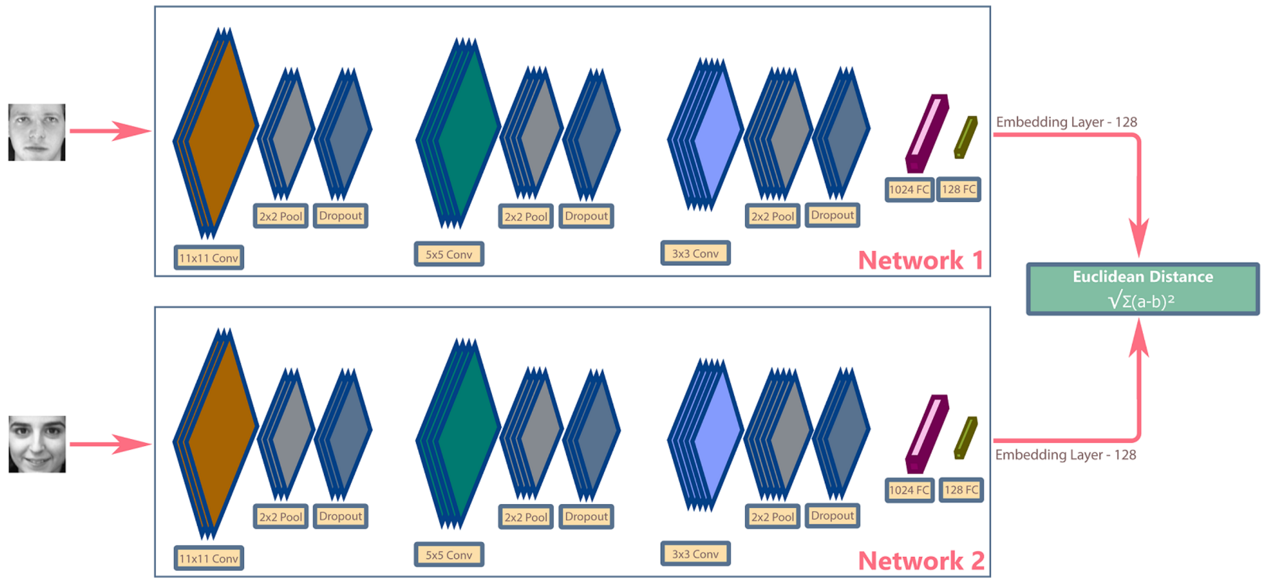


Figure 4: Architecture of a Siamese Network.

The two networks are identical, as shown in the above diagram. The Siamese Network functions in the following manner.

- A pair of images are selected from the dataset and processed by one of the networks above to train a Siamese Network.
- Because the networks have the same structure, the same processes will be performed on the corresponding images.
- Finally, the Neural Networks have Fully Connected Layers, the final of which has 128 nodes. When the network is applied to the image, this layer is the final feature that appears. The Embedding Layer Representation is what it's called. As a result, the Siamese Network produces two separate Embedding Layer Representations for the two images in the pair.
- Following that, the Network calculates the Euclidean distance between the two embedding layers. If the images are of the same person, the embeddings should be highly comparable, therefore the distance between them should be reduced. If the images are of different persons, though, the distance is likely to be greater.
- The distance value is transformed into a 0–1 range using a Sigmoid Function.
- A loss function is applied to the sigmoid result to penalise the network for updating its weights and biases. The weights and biases are updated in the same way on both networks.

This process repeats for all the image pairs generated from the dataset.

## 4 Dataset

Performance of a machine learning algorithm partly depends on the data. With the increasing amount of data, a machine learning algorithm performs well. Training a model with large amount of data helps to build a powerful model.

In our project, we have used labeled data for training the model in order to implement supervised learning. We have used Labeled Faces in the Wild (LFW) and Correctly Masked Face Dataset (CMFD) data set in our work.

1. **Labeled Faces in the Wild (LFW):**
   This dataset is consisting of photographs of faces which popular for unconstrained face recognition related studies. This was created and maintained by researchers at the University of Massachusetts, Amherst [6]. This dataset has 13,233 images of 5,749 people collected from the web. The resolution of the images are 250 x 250.

Figure 5: Labeled Faces in the Wild

2. **Correctly Masked Face Dataset (CMFD):**
   This data set is consisting of correctly and incorrectly worn masks on human faces. These masks were added artificially in the faces. There are about 133,783 images in the dataset [1]. This dataset is popular for the study of facial mask related problem.

Figure 6: Correctly Masked Face Dataset

In order to build a facial mask detection and facial identification system, first, we need to train our model so that it can distinguish if a given image contains a human face or not. For that we have used a dataset which is a combination of noisy images and Labeled Faces in the Wild.

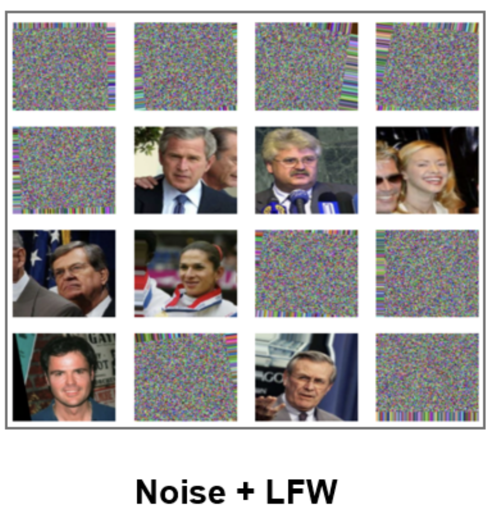


Figure 7: Combination of noise and LFW dataset

In order to train our model to detect if a given image has face with facial mask or not, we have combined noisy images and Correctly Masked Face Dataset and created a new dataset.From Correctly Masked Face Dataset, have only used the faces with correctly worn masks images.

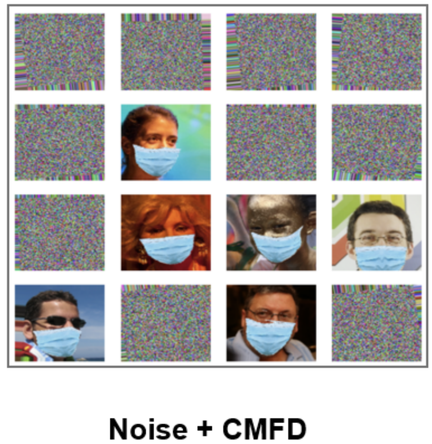


Figure 8: Combination of noise and CMFD dataset

Lastly, to build our final data set for the classification we will add both Labeled Faces in the Wild and Correctly Masked Face Datasets which will include faces with facial masks and faces without facial masks. Like before, from Correctly Masked Face Dataset, have only used the faces with correctly worn masks.


Figure 9: Combination of LFW and CMFD dataset

## 5 Data Pre-Processing:

In machine learning approach, it is very important to pre-process the data. It helps to improve the quality of data. Moreover, a model performs well if it is trained with cleaner and more organized dataset. For our work, we have used resize method as a learning algorithm works fast with smaller sized data. We have resized our images to 32 x 32.

In our datasets there are variety of images. Some could be from high pixel range and some could be from low pixel range. Therefore, it is very important to treat all the images in same scale. And so, we have rescaled the data by 1./255 to transform every pixel value from range [0,255] to [0,1].

In order to evaluate the performance of our model, we require validation set after training the model with train data. Hence, we have used validation split 0.2 and created training data and validation data from our dataset by splitting. As experimental setup we used Apple M1 chip MacBook Air with 8GB RAM and 256GB SSD. For implementation, we have used Keras and tensorflow as deep learning framework. Other than that, we have also used matplotlib, opencv, such open source libraries.

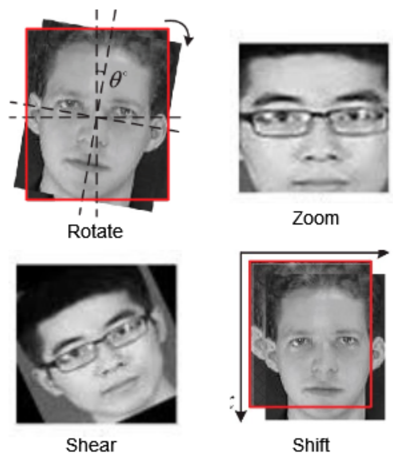


Figure 10: Data Augmentation methods

As mentioned earlier, machine learning algorithm performs well with increasing amount of data, we have implemented several data augmentation methods. This has helped us to train a model with large amount of data for better performance. The used data augmentation methods are-

1. **Rotation:**
   We can create new images by rotating existing images. We have rotated images by 10 degrees to create more new images for our datasets.

2. **Shear:**
   We can also create more data using shearing. It uses technique of shifting one part of the image like a parallelogram. We have used shear range 0.2.

3. **Zoom:**
   Zooming an image and using that zoomed image as a new image is another way of data augmentation. We have used zoom range 0.1 for our datasets.

4. **Shift:**
   A shift to an image involves moving all pixels of the image in one direction, such as horizontally or vertically, while keeping the image dimensions the same. We have used both horizontal and vertical shift ranging 0.1.

## 6 Proposed Model

In this approach, we have built a Convolutional neural network (CNN) model to identify faces with facial masks and faces without facial masks. A machine learning package, Keras has been used to build the CNN model. Our model consists of multi-convolutional (6 convolutional layers) with different amount of 3 x 3 filters to build a high performing model. As activation function, we have used Relu as it is a popular choice for machine learning approaches for its outstanding result. We have used 6 max pooling layers as well with 2 x 2 pool size with stride of 2. We have also added dropout layers in our model and kept the dropout rates higher in those layers where number of parameters were higher. The description of each layer of our CNN model is given below:

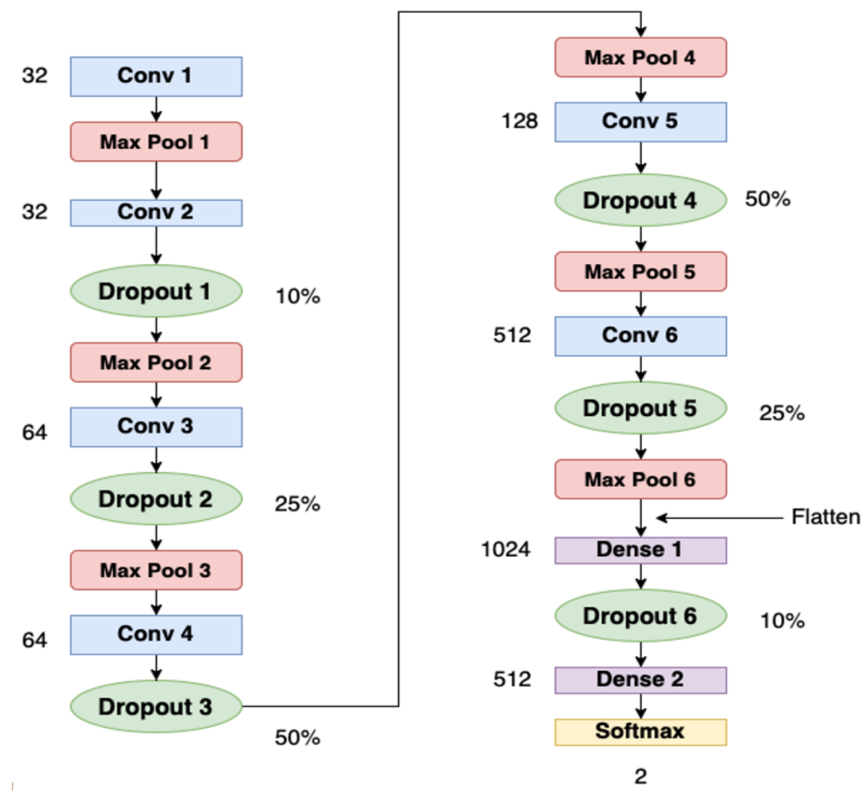


Figure 11: Proposed CNN model

For the first convolutional layer, we have used 32, 3 x 3 filters for 32 x 32 RGB inputs. We kept the padding same. Our first max pooling level has 2 x 2 pool size with stride 2 keeping the padding same.

In the second convolutional layer, we have used 32, 3 x 3 filters keeping the bias true and padding same. We added out first dropout layer with dropout rate 0.1. Our second max pooling level also has 2 x 2 pool size with stride 2 and padding has been kept valid.
The third convolutional layer has 64, 3 x 3 filters and other attributes were kept same as before. For this layer, we have increased the dropout rate to 0.25. The third max pooling layer was

kept the same as before.

The fourth convolutional layer is identical to the third convolutional layer. We have again increased the dropout rate to 0.50. The fourth max pooling layer is same as before.

In the next convolutional layer, 128, 3x3 filter has been used and other parameters were the same as before. Both dropout rate and the next max pooling layer was same like before.

The final convolutional layer was consisting of 512, 3x3 filters. For this layer, we decreased the dropout rate to 0.25. And the last max pooling layer was also consisting of 2 x 2 pool size with stride 2 keeping the padding same.

Then a flatten layer is added to reshape the tensor. At the end we have included 2 dense layers, one with 1024 neurons and another with 512 neurons. We have used Relu as activation function for these dense layers keeping the bias true. An intermediate dropout layer has also been added with 0.1 dropout rate. Finally, a output layer has been added with two units with a SoftMax activation function.

For this model we have used Adam optimizer with learning rate 0.0005. the used loss function is Sparse categorical cross entropy and metrics is accuracy.

Table 1: Accuracy, Precision, Recall and F1 scores for each datasets

| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|
| Face + Noise | 100% | 100% | 100% | 100% |
| Masked Face + Noise | 100% | 100% | 100% | 100% |
| Face + Masked Face (Non-tuned) | 43% | 22% | 50% | 30% |
| Face + Masked Face (tuned) | 100% | 100% | 100% | 100% |

# 7 Results and discussion

As a part of our experiment, we wanted to get a good sense of how transfer learning works. So, we subdivided our broader problem into smallest part, worked with that, and later extended it to our final problem. The following steps explain our approach in detail:

## 7.1 Step-1: Face Detection, Model-1:

At this stage, we used our proposed model to distinguish between face images of LFW dataset and noisy images. After training the model for 30 epoch it gave and excellent accuracy, precision, recall and F1-score of 100% (Table 1). We can also see the confusion matrix and, learning curves for test and validation accuracy and loss in Fig. 12. However, after training the model for face recognition, we saved its weights to transfer to recognize only masked face.

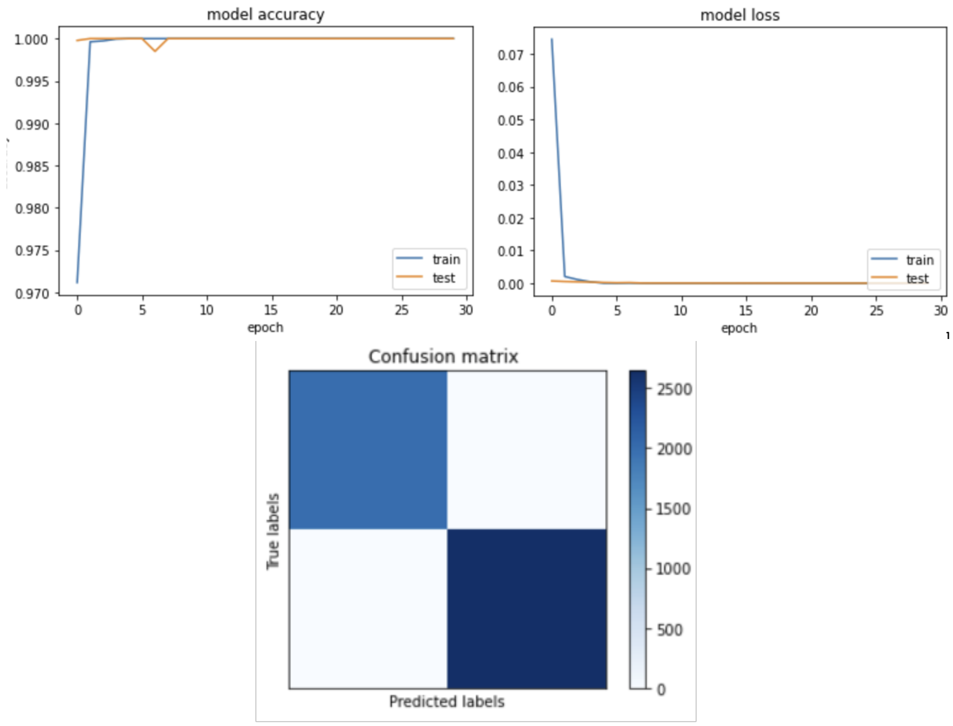


Figure 12: top-left: accuracy vs epochs, top-right: loss vs epoch, bottom: confusion matrix

### 7.2 Step-2: Masked Face Detection- transfer from face, Model-2:

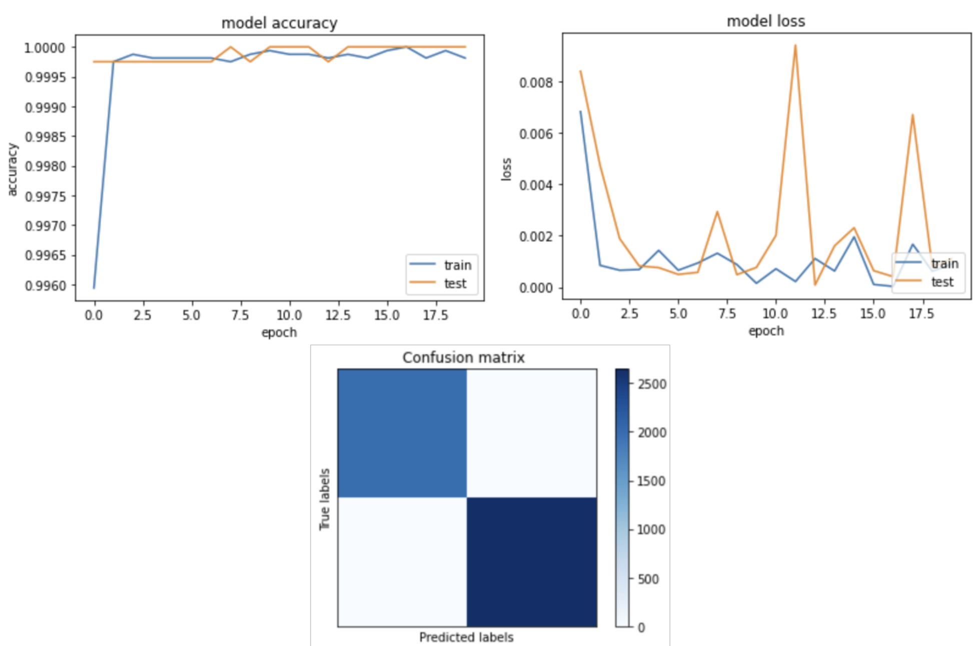


Figure 13: top-left: accuracy vs epochs, top-right: loss vs epoch, bottom: confusion matrix

At this stage we used Model-1 as our base model for transfer learning. As transfer learning setting, we freeze all the layers except the last two. This model-2 was running for 20 epochs.

And the recognition for this classification remains the same as before, 100%. We can see the associated diagrams in Fig-13.

### 7.3 Step-3: Face and Masked Face Detection- transfer from face, Model-3:

Here we used our pre-trained Model-1 as base model as well for face vs masked face recognition task. Initially, we keep all the layers frozen except the last two and train the model for 15 epochs. The results were not good in this case. The classifier identified all the images as one category, as a result, the overall accuracy was 43%, precision of 22%, recall of 50% and F1-score was 30%. The confusion matrix and the learning curves are shown in Fig-14. However, due to the lack of performance we tuned our base-model, Model-1 in the next stage.

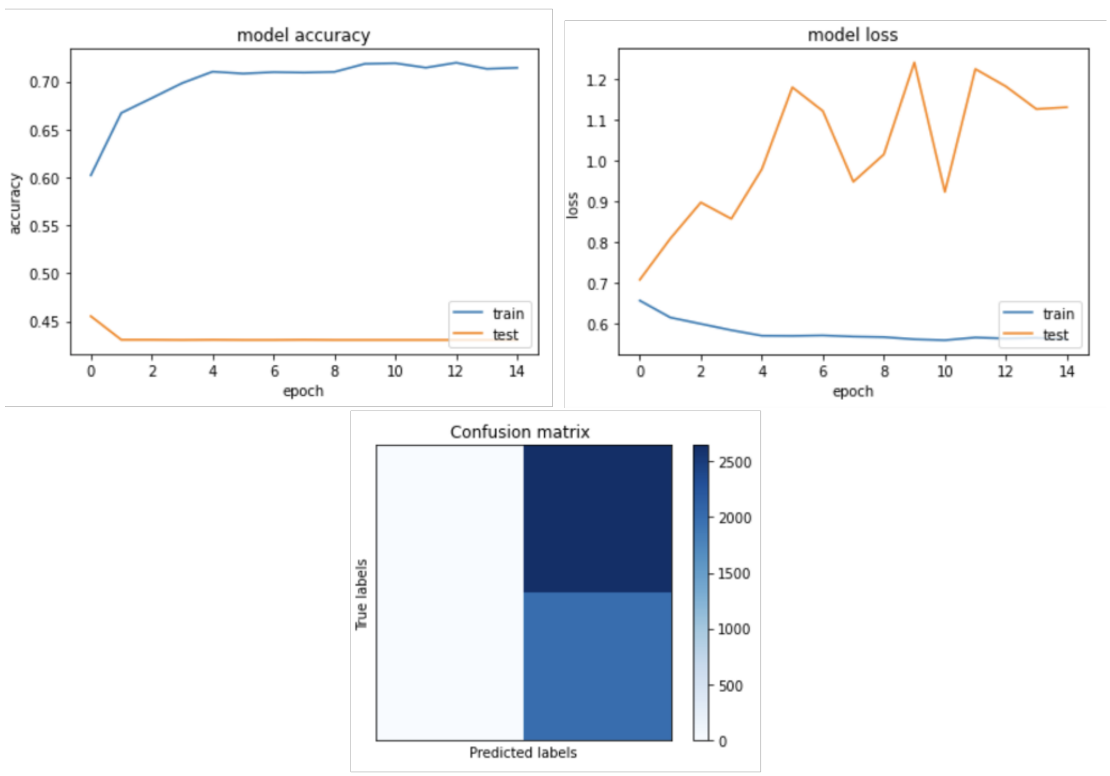


Figure 14: top-left: accuracy vs epochs, top-right: loss vs epoch, bottom: confusion matrix

### 7.4 Step-4: Face and Masked Face Recognition- transfer from face (tuned), Model-4:

failed, this time we freeze only the even layer's weights. After training for 15 epochs with face vs masked face dataset, we got excellent accuracy, precision, recall and F1-score of 100%. We can see the learning curves and confusion matrix for this model in Fig- 15. Now that we have our final model, Model-4, which can identify masked faces and non-masked faces, we can move on to the next phase of our project of face identification.

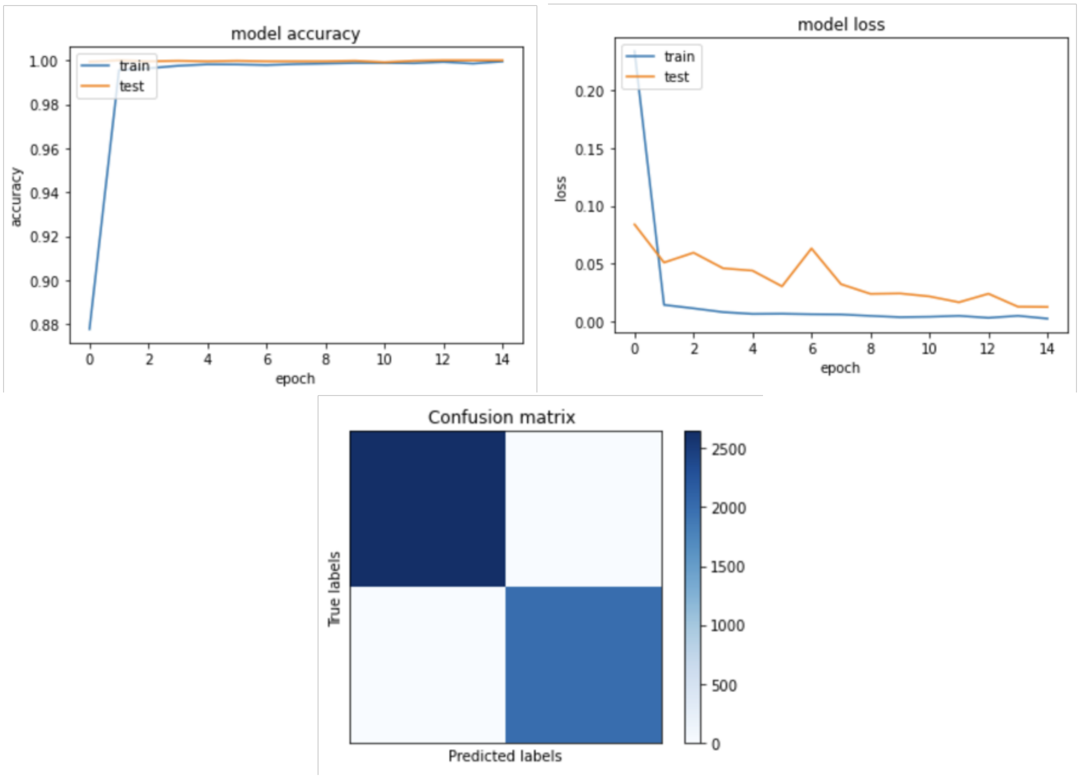


Figure 15: top-left: accuracy vs epochs, top-right: loss vs epoch, bottom: confusion matrix

### 7.5 Step-5: Face identification:

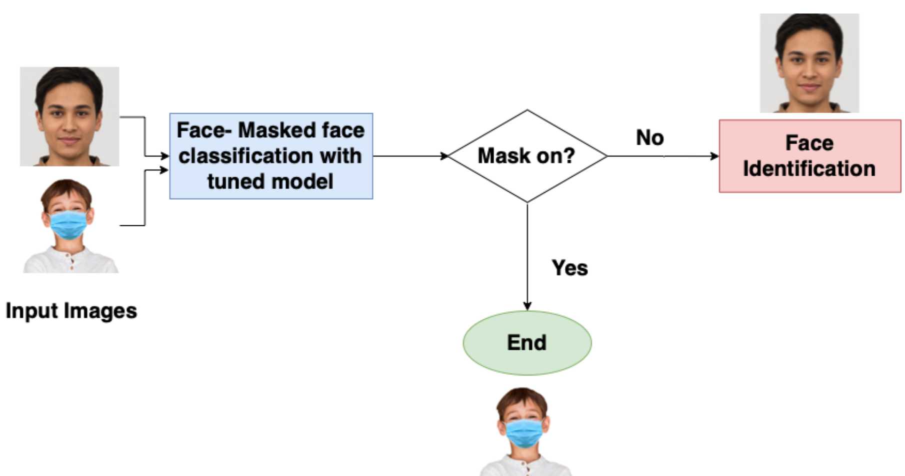


Figure 16: Overall System Diagram

We have our final model which can correctly identify face and masked faces. The images of detected faces without masks will now be used by our face identifier. A detailed view of our overall system is depicted in Fig-16. To implement the face identification system, we have made use of the concept of Siamese network. We complied our trained final Model-4 with triplet loss function and adam optimizer with learning rate 0.0005. The last two layers of this model were removed, so now, when we give an image input to the model, it outputs a 1024 size of embedding for that given image. Initially, we store the embeddings of some known person in a database. Later, whenever we got an unmasked face, we calculate and then compare it's embedding with the embeddings of the database. If the embeddings are similar, we can then link the persons' name with the unmasked face. An example output from our model along with some other details is shown on Fig-17. In this example case, our model identified the unmasked persons' name as being 'Aron Eckhart'.

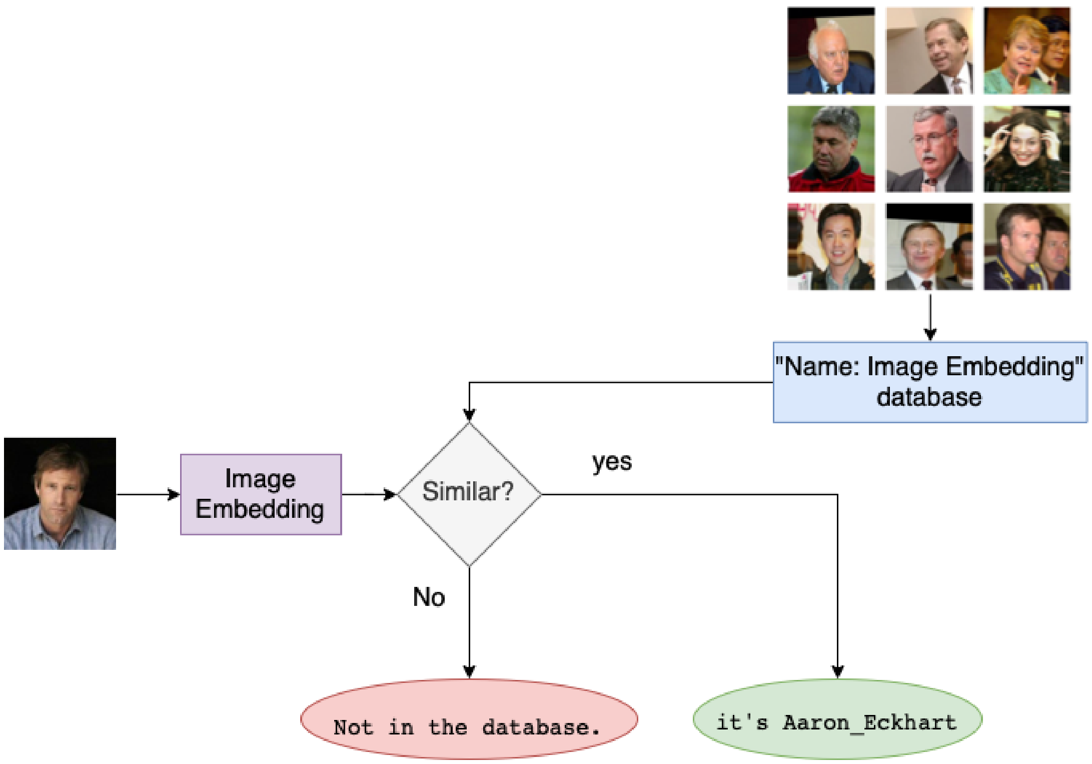


Figure 17: Example Identification by our proposed system

## 8 Conclusion and Future works

In this project, we have demonstrated a system which will be able to detect people who don't wear face masks and able to detect those people's identity. In this stage of COVID-19 spread, this automated system will be very helpful to monitor if safety measurement followed properly. We have implemented Convolutional neural networks with transfer learning and Siamese network to build this system. The test result showed a high accuracy performance in detecting faces without facial masks. Our model was able to achieve 100% accuracy rate for facial mask detection.

In this project, we have worked with images to built this model. In future, we can work with video data to detect face without mask and identify those individuals. With this approach, we can implement this system in CCTV cameras to provide a more versatile monitoring system. Moreover, we can collect more data from different distribution for making this model more powerful. Furthermore, if we can manage individuals contact information, we can extend this system to build a system which will be able to send alert to the individuals without facial mask worn. With this view, this system can help to ensure a smart monitoring system for maintaining safety measurements for COVID nation-wide.

## References


[1] H. Benhabiles A. Cabani, K. Hammoudi and M. Melkemi. Maskedface-net - a dataset of correctly/incorrectly masked face images in the context of covid-19. *Smart Health, ISSN 2352-6483, Elsevier*, https://doi.org/10.1016/j.smhl.2020.100144, 2020.

[2] Mira M. Boulos. Facial recognition and face mask detection using machine learning techniques. *Theses, Dissertations and Culminating Projects*, 728, 2021.

[3] Musab Coşkun, Ayşegül Uçar, Özal Yildirim, and Yakup Demir. Face recognition based on convolutional neural network. pages 376–379, 2017.

[4] Wasif Ansari M. Das, A. and R. Basak. Covid-19 face mask detection using tensorflow, keras and opencv. *IEEE 17th India Council International Conference (INDICON).*, DOI: 10.1109/INDICON49873.2020.9342585, 2020.

[5] M. Farayola and A. Dureja. A proposed framework: Face recognition with deep learning. *International Journal of Scientific  Technology Research.*, Volume 9, Issue 7, ISSN 2277-8616, 2020.

[6] T. Berg E. Learned-Miller G. B. Huang, M. Ramesh. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report 07-49*, 2019.

[7] H. Ge, Y. Dai, Z. Zhu, and B. Wang. Robust face recognition based on multi-task convolutional neural network. *Mathematical Biosciences and Engineering*, 18(5):6638–6651, 2021.

[8] L. R. Garcia Godoy et al. Facial protection for healthcare workers during pandemics: a scoping review, bmj glob heal. Article e002553, 10.1136/bmjgh-2020-002553,(5 (5) (2020)).

[9] R. J. Hassan and A. Abdulazeez. Deep learning convolutional neural network for face recognition: A review. *International Journal of Science and Business*, 5(2):114–127, 2021.

[10] Mohsen Heidari and Kazim Fouladi-Ghaleh. Using siamese networks with transfer learning for face recognition on small-samples datasets. pages 1–4, 2020.

[11] Zhenye Li, Hongyan Zou, Xinyan Sun, Tingting Zhu, and Chao Ni. 3d expression-invariant face verification based on transfer learning and siamese network for small sample size. *Electronics*, 10(17), 2021.

[12] N. K. Mishra and S. Kumar Singh. Face recognition using 3d cnns. 10.1007/978-981-16-1681-5, 2021.

[13] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth. Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2. *Sustainable Cities and Society*, 66(10269):2, 2021.

[14] Thenmoezhi N. Gayathri M. Prakash, R. M. Face recognition with convolutional neural network and transfer learning. *International Conference on Smart Systems and Inventive Technology (ICSSIT).*, DOI: 10.1109/ICSSIT46314.2019.8987899, 2019.

[15] S. Sethi, M. Kathuria, and T. Kaushik. Face mask detection using deep learning: An approach to reduce risk of coronavirus spread. *Journal of Biomedical Informatics*, 120(10384):8, 2021.

[16] F. M. Shamrat, Sovon Chakraborty, Md Billah, Md Jubair, Md Saidul Islam, and Rumesh Ranjan. Face mask detection using convolutional neural network (cnn) to reduce the spread of covid-19. 10.1109/ICOEI51242.2021.9452836., 2021.

[17] K. H. Teoh, R. C. Ismail, Szm Naziri, R. Hussin, Mnm Isa, and Mssm Basir. Face recognition and identification using deep learning approach. *Journal of Physics: Conference Series*, Volume 1755 Ser. 1755 012006.

[18] Ayşegül Uçar, Yakup Demir, and Cüneyt Güzeliş. Object recognition and detection with deep learning for autonomous driving applications. *SIMULATION*, 93(9):759–769, 2017.

[19] Yu H. Wang D. Li G. Wang, D. Face recognition system based on cnn. *International Conference on Computer Information and Big Data Applications (CIBDA).*, DOI: 10.1109/CIBDA50819.2020.00111, 2020.

[20] Y. Li X. Zhao Y. Li, Z. Wang and H. Huang. Design of face recognition system based on cnn. *Journal of Physics: Conference Series.*, 1601. 052011. 10.1088/1742-6596/1601/5/052011, 2020.

[21] Li Yao. A compressed deep convolutional neural networks for face recognition. pages 144–149, 2020.

[22] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[23] Jianming Zhang, Xiaokang Jin, Y. Liu, A.K. Sangaiah, and J. Wang. Small sample face recognition algorithm based on novel siamese network. *Journal of Information Processing Systems*, 14:1464–1479, 01 2018.