НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
май–июнь 2022 Том 22 № 3 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
May–June 2022 Vol. 22 No 3 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

doi: 10.17586/2226-1494-2022-22-3-528-537

Cloud-based intelligent monitoring system to implement mask violation detection and alert simulation

Komal Venugopal Vattumilli[1]✉, Lalith Movva[2], Arun Kumar Thangavelu[3], Jayashree Jayaraman[4], Vijayashree Jayaraman[5]

[1,2,3,4,5] School of Computer Science and Engineering, Vellore Technological Institute, Vellore, 632014, India

[1] komalvenugopal@gmail.com✉, https://orcid.org/0000-0002-9292-7473
[2] lalithindian@gmail.com, https://orcid.org/0000-0002-1767-3186
[3] arunkumart@vit.ac.in, https://orcid.org/0000-0001-7384-8975
[4] jayashree.j@vit.ac.in, https://orcid.org/0000-0002-0191-8070
[5] vijayashree.j@vit.ac.in, https://orcid.org/0000-0001-6987-3377

Abstract
The importance of wearing a mask in public places came to light when the COVID-19 pandemic has started due to the coronavirus. To strictly control the spread of the virus, wearing a mask is mandatory to avoid getting the virus through others or spreading the virus to others if we are carrying it. Since it's not possible to check each individual in public places whether he/she is wearing a mask, this paper proposed a face mask detection using Deep Learning (DL) and Convolutional Neural Network (CNN) techniques. A cloud-based approach that adopted DL is used to identify the persons violating the rules. The dataset used in the work is collected from various studies, such as Prajnasb/observations and Kaggle's Face Mask Detection Dataset that contains images of people wearing and not wearing masks. The faces in the images will be detected and cropped with the help of a trained face detector which will be used for checking whether the face in the image is wearing a mask or not. Face mask detection is done with the help of CNN. The input image is fed into the CNN and the output is binary format, whether person wearing or not wearing a mask. The work uses Max Pooling and Average Pooling layers of CNN. The outcome of the work shows that the proposed method achieves 98 % of accuracy using Max Pooling which is better than the currently available works.

Keywords
convolutional neural networks, CNN, PyTorch, deep learning, cloud

For citation: Komal Venugopal V., Lalith M., Arun Kumar T., Jayashree J., Vijayashree J. Cloud-based intelligent monitoring system to implement mask violation detection and alert simulation. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2022, vol. 22, no. 3, pp. 528–537. doi: 10.17586/2226-1494-2022-22-3-528-537

УДК 004.932.2

Облачная интеллектуальная система мониторинга для обнаружения нарушений ношения маски и выдачи предупреждений

Ваттумилли Комал Венугопал[1]✉, Мовва Лалит[2], Тангавелу Арун Кумар[3], Джаяраман Джаяшри[4], Джаяраман Виджаяшри[5]

[1,2,3,4,5] Школа компьютерных наук и инженерии, Технологический институт Веллору, Веллуру, 632014, Индия

[1] komalvenugopal@gmail.com✉, https://orcid.org/0000-0002-9292-7473
[2] lalithindian@gmail.com, https://orcid.org/0000-0002-1767-3186
[3] arunkumart@vit.ac.in, https://orcid.org/0000-0001-7384-8975
[4] jayashree.j@vit.ac.in, https://orcid.org/0000-0002-0191-8070
[5] vijayashree.j@vit.ac.in, https://orcid.org/0000-0001-6987-3377

© Komal Venugopal V., Lalith M., Arun Kumar T., Jayashree J., Vijayashre J., 2022

Аннотация

Важность ношения маски в общественных местах стала очевидна, когда из-за коронавируса началась пандемия COVID-19. Для строгого контроля за распространением вируса ношение маски является обязательным. В общественных местах нет возможности проверить каждого человека на ношение маски. Предложен способ обнаружения маски на лице человека с помощью методов глубокого обучения и сверточных нейронных сетей (Convolutional Neural Network, CNN). Облачный подход основан на глубоком обучении и используется для выявления лиц, нарушающих правила. Набор данных, примененный в работе, заимствован из научных исследований, таких как Prajnasb/observations и набор данных Kaggle Face Mask Detection. Данные содержат изображения людей в масках и без них. Лица на изображениях обнаруживаются и выделяются с помощью специального обученного детектора лиц. Обнаружение маски осуществлено с помощью детектора и CNN. Входное изображение направляется в сеть, а выходные данные представляются в двоичном формате независимо от того, обнаружена маска или нет. В работе использованы слои CNN Max Pooling и Average Pooling. Результат исследования предложенного метода показал, что метод позволяет достичь 98 % точности на максимальном объеме тестовых изображений.

Ключевые слова

сверточные нейронные сети, CNN, PyTorch, глубокое обучение, облако

Ссылка для цитирования: Комал Венугопал В., Лалит М., Арун Кумар Т., Джаяшри Д., Виджаяшри Д. Облачная интеллектуальная система мониторинга для обнаружения нарушений ношения маски и выдачи предупреждений // Научно-технический вестник информационных технологий, механики и оптики. 2022. Т. 22, № 3. С. 528–537 (на англ. яз.). doi: 10.17586/2226-1494-2022-22-3-528-537

Introduction

The COVID-19 pandemic has created a major impact on the world in many ways such as affecting the economy and also stopped people for many days who has to travel from one place to another place. Since the virus transmits from one person to another through expired air, the speed of the transmission is increasing rapidly [1]. Having the need to stop the spread of the virus, which is increasing day by day, many governments all across the world have imposed lock down, and they allow people to come out only to buy daily essentials. But there also exists a high risk of coming out from the houses at the contaminated zones since the spread of the virus is very high in those zones [2]. To prevent the spread of the virus, many countries have imposed strict rules about wearing a mask in public places. Since all the people cannot be proctored individually to check whether they are wearing a mask or not, with the help of artificial intelligence we can get an analysis about the places where most of the people are not wearing masks with the help of deep CNN. This model can be implemented in real time cameras of Closed Circuit Television (CCTV), and there are no restrictions since everything is cloudified and can run anywhere because there is no platform dependency in any step.

DL models are used recently to learn the features of the dataset which help in handling the large amount of data that we get from different sources. It is almost impossible to classify this data using normal Machine Learning Algorithms. The classical Machine Learning Algorithms learn the data by parsing it which leads to poor accuracy when we have diverse and huge data. In case of this type, DL helps by creating an artificial neural network that learns from previous experiences without getting programmed explicitly [3]. DL procedure creates its own decisions and doesn’t depend only on the data provided. There are different types of DL techniques available: multi-layer perception, Recurrent Neural Networks, CNN, and Modular Neural Networks which have their use-cases.

CNN is the popular class in Learning for its ability to understand visual imagery. CNN have been a revolution on the computer vision domain. Initially, there were some image processing techniques but they were not able to generate a state-of-the-art result. Whenever CNN have been introduced, they have started producing state-of-the-art results [4]. A specified kernel is placed on the image matrix, and a new convolution matrix is formed from the combination of these two matrices. Pooling is applied on the newly formed matrix that helps in reducing the dimensions of the image so that the image processing gets done quicker than the previous one. In this paper, we use Max and Average Pooling layers in CNN and compare them with the results. Max Pooling is done by using the maximum pixel value when Kernel and Average Pooling takes the average pixel value. The image gets flattened after pooling and gets classified. This process gets repeated for a certain number of epochs for all the images. The complexity and size of DL models are drastically increasing to achieve state-of-the-art results. New model architectures are constantly being developed with millions and billions of learnable parameters in the model. Recently, OpenAI company has developed GPT-3 which consists of 175 billion parameters, and it has achieved state-of-the-art results at many tasks [5]. In the real-world scenarios, where we need to deploy these huge models, we might face issues such as storage, hardware requirements, etc., since these models require a lot of storage and advanced hardware for faster inference. We have used Knowledge distillation to reduce the size of the model by having a lesser number of parameters when compared to the original model; we are reducing the original model size by 40 % of the originally trained model such that the inference becomes 60 % faster and with the same accuracy as the Teacher model. The remainder of the paper is organized as follows.

We discuss the various studies done on the facemask detection using different algorithms in the recent years along with the results obtained. Later we give an overview of CNN Architecture; a dataset was used and it shows the working of our model with the detailed architecture diagram. Then we explain the metrics used to evaluate the performance of the model and compare the results to find the best accurate model. And finally, we outline the advantages of using the Cloud Architecture, the future scope of this work concluding the paper.

Literature review

In this section, the studies of different algorithms used in the currently available works are discussed. Fu et al. [3] proposed a fast detection framework /DSSD (Deconvolution Single Shot Detector). The work uses VOC and COCO datasets to evaluate the performance of their own model. They used pretrained ILSVRC CLS-LOC dataset for experimental evaluation. The work achieves overall 82 % of accuracy which is not efficient for proving the achievements. However, the work achieves training process in a less amount of time. Therefore, the time complexity of the training process is reduced compared to other works. CNN is the popular class in Learning for its ability to understand visual imagery. CNN have been a revolution on the computer vision domain. Initially, there were some image processing techniques as they were not able to generate a state-of-the-art result; but when CNN have been introduced, they have started producing state-of-the-art results. Lin et al. [4] proposed an object detection technique based on feature pyramids. A specified Kernel is placed on the image matrix and a new convolution matrix is formed from the combination of these two matrices. Pooling is applied on the newly formed matrix that helps in reducing the dimensions of the image so that the image processing gets done quicker than the previous one. The work uses COCO dataset for the evaluation of the proposed model. Author concludes that the Feature Pyramid Network is efficient for applying feature extraction based on CNN model. However, the training process is very difficult which leads to increased cost and time. Farfade et al. [6] describe the concept of extracting the faces from the given images, and it has been well explained in this paper. There has been specified how to extract the region of interest (in this case, it is the face of the person in the image), and we get the coordinates of the face in the image. The computational power required to train this model is very high, so it cannot be done in a CPU since there are many calculations to be done in parallel to save time; but computing with the help of CPU consumes a lot of time whereas while using a GPU it can be done very fast when compared to computational time using a CPU due to its parallel processing capability. Chen et al. [7] had compared the results between SVM, DNN, CNN, and CNN using transfer learning, and the CNN model with transfer learning and generated better results when compared to other models. Geoffrey et al. [8] proposed a Knowledge distillation model that is considered as a single model based on compression technique. Authors have suggested that the combination of difference machine learning algorithms to build a prediction is an inefficient approach that makes the process slow. Even though, the idea is quite different and useful, but the method cannot guarantee the better performance in terms of accuracy and time.

From the literature studies, all model focused only on the complexity of training which do not guarantee the better performance. We also observed that the robust Cloud Solution for the face-mask-detection is not available in the current scenario. The models used for identification are not so accurate. In this paper, we use Max and Average Pooling layers in CNN and compare them with the results. Max Pooling is done by using the maximum pixel value when Kernel and Average Pooling takes the average pixel value. The image gets flattened after pooling and gets classified. This process gets repeated for a certain number of epochs for all the images. We also adopted the approach proposed by Knowledge distillation [9] and compared the results for Max Pooling and Min Pooling of the Teacher model, and used the most accurate result in the Student model.

Proposed methodology

To build a prediction model, all the resources are created that are required for the flow of the process initially. Dataset details are given in the next section. The main objective of the paper is to improve the accuracy with reduced processing time. Amazon Web Services (AWS) are offered by Amazon with a wide range of services that are required for the end-to-end setup for any online application. AWS Lamda will help in executions of certain tasks and needs to be run after particular process completes its execution. In this model, we use the AWS Lambda resource which will help in running an event after it found any changes in the specified AWS Simple Cloud Storage (AWS S3) location. Amazon Elastic Compute Cloud (AWS EC2) instance is a remote machine that can act as the server and is used to run all the scripts and maintain the MYSQL database. AWS S3 is a remote storage resource that helps in storing the videos and images of the violators and processing them to the server. Amazon Simple Notification Service (AWS SNS) helps in transferring mails and alerts to the subscribed users from AWS Lambda. AWS Identitity access management (IAM) is used in creating login username and password for different users. We need to attach the permissions required for the operations to that user. These permissions are attached as a json code which is called a policy. As a first part of the work, a role is created in AWS IAM with the policy attached to it, having all the access permissions for the two AWS S3 buckets, AWS Lambda, AWS EC2, and AWS SNS. The login credentials are generated from Security Credentials in IAM and the same role can be used across every stage of the process. We can communicate to AWS cloud by configuring the AWS credentials in AWS EC2 within the default profile. An SNS subscription has to be created which will mail to the email-id provided after AWS Lambda is invoked. In this proposed model, the OpenCV records the video of the surroundings every 4 hours. The video is then uploaded into the first AWS S3 bucket by configuring the AWS Access credentials using the Python Boto3 package having the city name as the prefix in the S3 bucket. We can process multiple videos from multiple locations that get forwarded to a folder in S3 bucket having the zip code of the location as a folder name [9]. Now AWS Lambda has to be set up to get data from AWS S3 and dump the data into an Amazon EC2 instance. AWS Lambda will apply SSH to the EC2 instance using the Python Paramiko library and copy the files from AWS S3 to the folder in EC2 instance. After copying the data, AWS Lambda will trigger a Python script that will run the

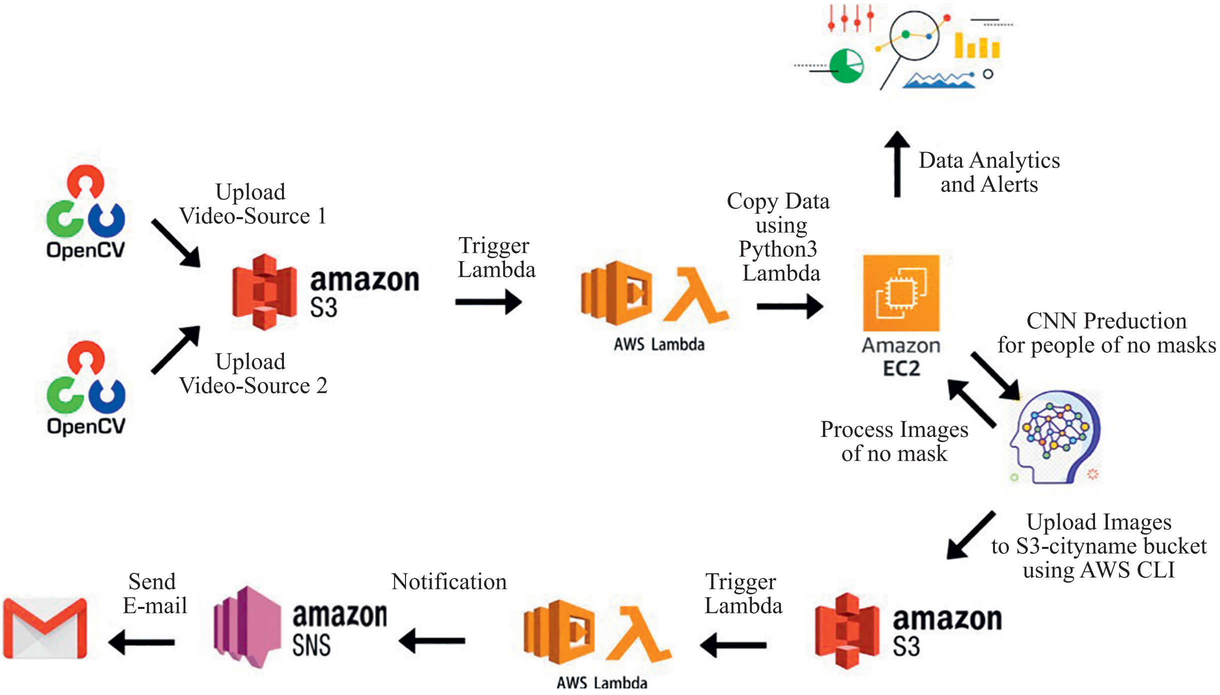
OpenCV
OpenCV
Upload Video-Source 1
Upload Video-Source 2
amazon S3
Trigger Lambda
AWS Lambda
Copy Data using Python3 Lambda
Amazon EC2
Data Analytics and Alerts
CNN Preduction for people of no masks
Process Images of no mask
Upload Images to S3-cityname bucket using AWS CLI
amazon S3
Trigger Lambda
AWS Lambda
Notification
amazon SNS
Send E-mail

Fig. 1. Architecture for Sending Mails of Violator Images

CNN model and identify the images of persons not wearing the masks. The images are then pushed into an MYSQL Database from the EC2 instance. When the image is pushed into MYSQL Database, it also gets pushed into the AWS S3 location. The push into the AWS S3 location will trigger an AWS Lambda function that will send the image of the violator (as in Fig. 1) to the mailing address mentioned in the AWS SNS Subscription.

Data set details

The dataset used in this paper is collected from different online resources available which included data from Kaggle's Face Mask Detection Dataset and Prajnasb/observarion dataset. There are two classes in this problem: people wearing masks and people not wearing masks. To avoid the problem of over fitting, the images were collected almost equal in number. The total number of images contained in the dataset is 7553 [10]. The number of images of the people who are wearing masks is 3725 and the number of images of the people who are not wearing masks is 3828. Prajbnasb dataset is helpful in the work to identify the facial features of person. This dataset consist of 1376 images: 690 of these with wearing mask and 686 — without wearing mask. Many features like nose, eyes, mouth, eye brows and many others are located using facial landmarks of the dataset. Sample images for dataset without mask and images for without mask are shown in Fig. 2 and 3.

In Fig. 2, facial landmarks are used to automatically gather the location of facial features such as chin, nose, and mouth. Fig. 2 represents the images of faces in different views and the system shows "no mask detection". If all features such as nose, mouth, and chin are visible, then the system detects "no mask".

Fig. 3 shows the faces of people who worn mask in different positions. In Fig. 3, *a*, *c*, the mask fits perfectly. Improperly worn mask images are shown in Fig. 3, *b*, *d*. If

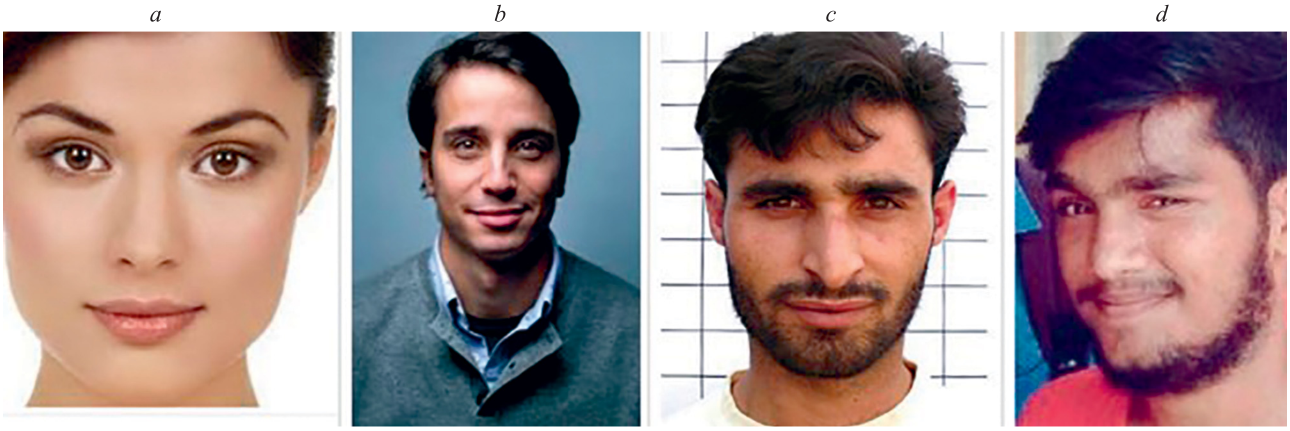

Fig. 2. Sample images belonging to dataset without mask

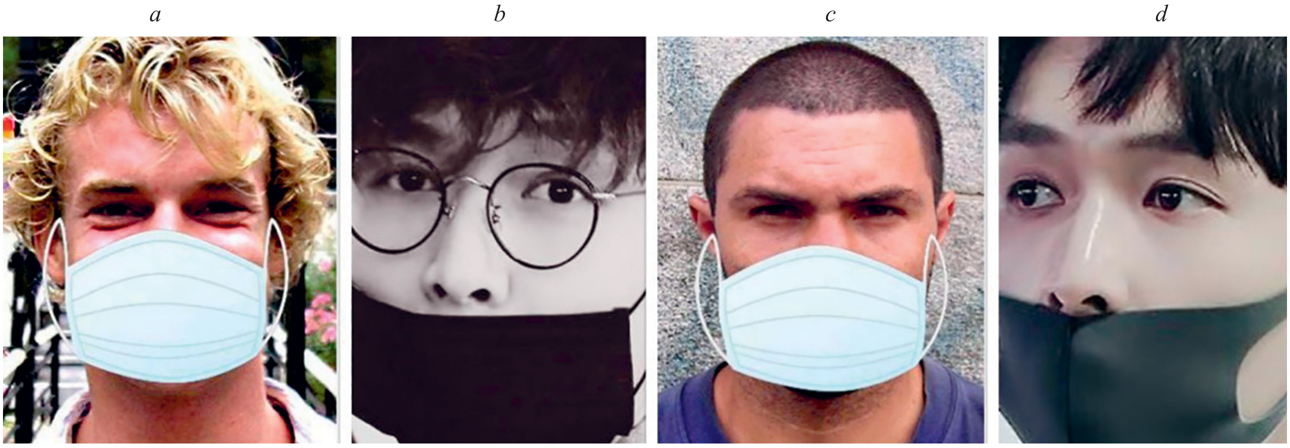

Fig. 3. Sample images belonging to data set with mask

any of the three features visible, then the system detects that the mask not worn properly.

Training the CNN Model

In this work, we have used DL Knowledge distillation and OpenCV library. The live video is captured by the cameras such as CCTV, and OpenCV accesses the live video. Then the video is split to different frames per second and the frames are fed to the pre-trained model that is implemented in this paper for inference to predict whether the person in the frame wears a mask or not. The result is highlighted in the video itself; the number of people in the frame might be more than 1. The frames are cropped in such a way that the faces of the people in the frame are detected using a pre-trained face detection model which gives us the coordinates of the face in the frame. The model to detect the face mask is trained in this work. We used CNN architecture to build the model as it is more suitable for vision tasks.

The CNN architecture in this work in Fig. 4 is made up of Convolutional and Max Pooling layers. To reduce the over-fitting in this model we have used some regularization techniques such as dropout which makes sure that no particular neuron plays a major role in the model while inferencing. First, the input image is pre-processed by using some augmentation techniques, and later it is converted to numeric matrices of 3 dimensions where each dimension contains the intensity values of red, green and blue in the pixel. Then these values are fed into the model as input. Second, the input values pass through the Convolutional layer where a filter of (3, 3) used, and padding and a stride of value 1 are applied on the input. Padding is applied to prevent the loss of information at the corners of the image so that the prediction might not be accurate if it misses some information. A ReLU activation function is used after the Convolutional layer. The inputs are passed to the ReLU activation function and then to Max Pooling layers where the dimension of the input gets reduced. A (3, 3) Kernel is used in Max Pooling, the dimension gets reduced and the important information is only preserved in the surrounding pixels. Then after passing the input data through a few Convolutional and Max Pooling layers, it is flattened and passed through a fully connected linear layer and then through a sigmoid layer where we get the output probability of the neuron. If the probability is above 0.5, then the person in the frame is wearing a mask; if it is less than 0.5, then the person in the frame is not wearing a mask. Since this comes under the binary classification problem, a binary cross-entropy loss is used here and the optimizer used for updating the weights is the Adam optimizer. 98 % of accuracy is observed using the Max Pooling in the CNN Architecture as shown in Fig. 5.

The CNN model using the Average Pooling has produced an accuracy of 97 % in detecting the facemasks.

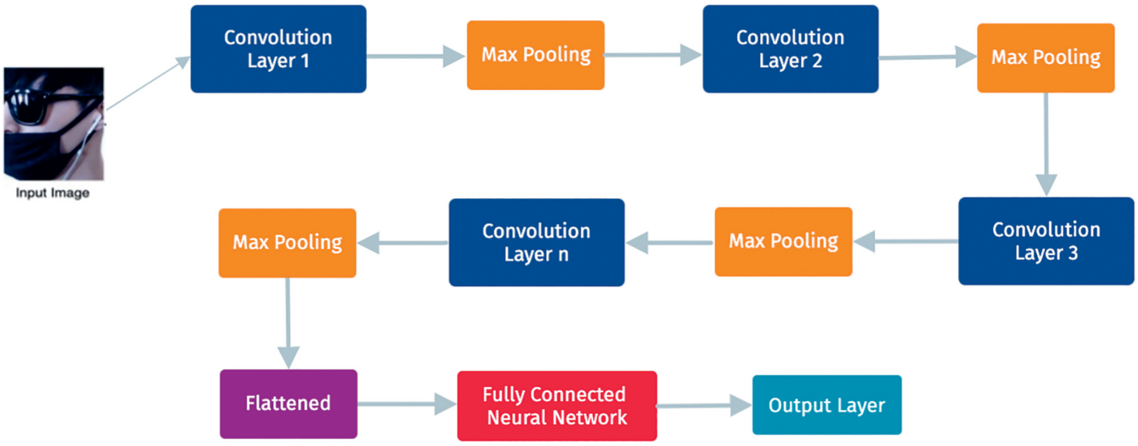

Fig. 4. CNN Architecture for Max Pooling

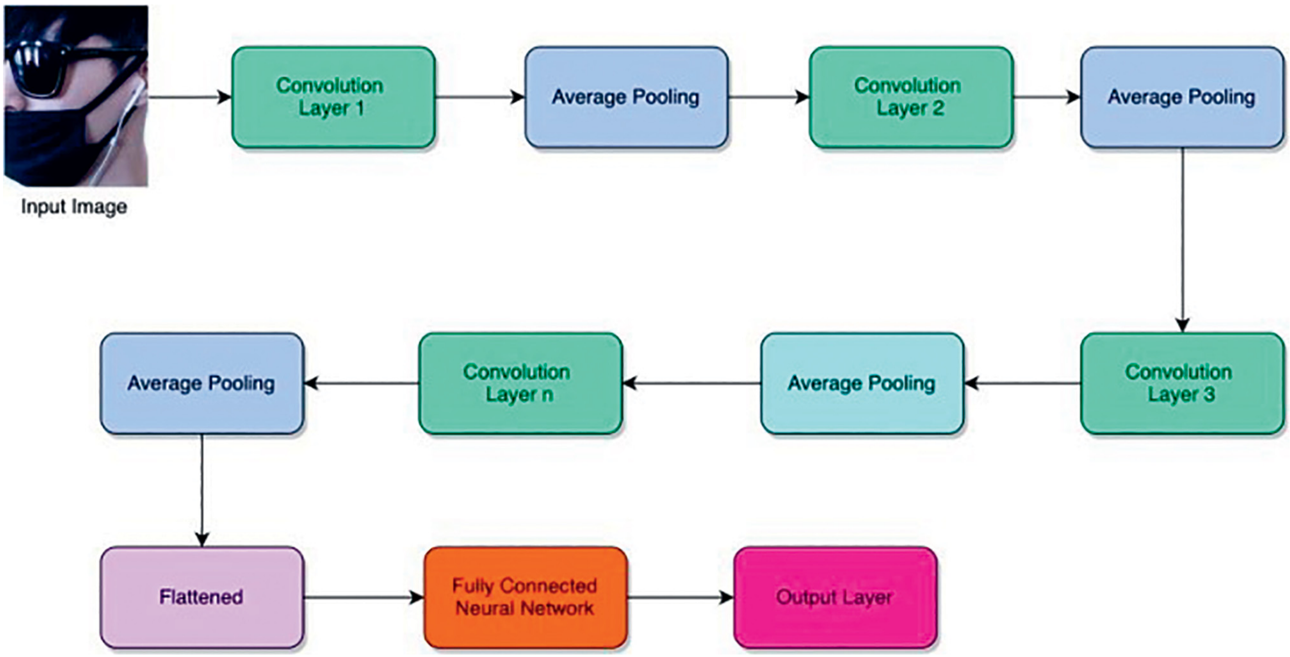

Fig. 5. CNN Architecture for Average Pooling

So, the Max Pooling was found to give the best results due to which it is used as Teacher model in the Knowledge distillation which is discussed in the next section.

Knowledge distillation

This model uses 6 convolutional and 3 dense layers containing a large number of trainable parameters. In this case, the inferencing becomes difficult because the input frames are continuously fed into the model and the output should be generated spontaneously [11]. But if we use such a large model trained, the inference becomes difficult, so we have used Knowledge distillation to overcome this problem. The model trained above is considered as Teacher model and the Student model consisting of lesser parameters than the trained Teacher model. The latter is trained using the training data for 50 epochs where the loss function in the Student model differs from the Teacher model. The loss function in the Student model is the sum of loss functions on hard labels and the loss on soft labels, which are the normalized probability distribution of the Teacher model output logits, calculated using temperature-based Softmax function. The parameters are shown in Table.

In the training, the temperature controls the peak distribution of the normalized probabilities. The peak values will become more peak and the least values will reduce further by applying temperature to the Softmax function. The parameters which are not learnable and defined manually are called hyperparameters. During the training, the learnable parameters get updated every iteration but there are no fixed values for hyperparameters. The hyperparameters, set while training the DL model, are the learning rate that is used in the optimizer, in the number of epochs the model is trained, in the batch size of the data while training, and in the activation functions used [11]. These hyperparameter values are experimental, so we have experimented with different values and finalized a model which achieves better accuracy with less over fitting. The temperature value in the Softmax function during Knowledge distillation is also a hyperparameter that varies from 0 to 1.

Saving and Loading Model

The model is trained with hyperparameters which produce better results. When the model is used in real-world scenarios, we can’t train the model every time on the device. So, we are saving the model in such a format so that it can be loaded whenever it is required. The weights freeze after the training and then are saved into a particular format such as h5 or pt based on the framework the model is trained on. In this case, we have used PyTorch for data augmentation and model training, then saving the model. When the model is used in a flask web framework, the model is loaded into the RAM when the service is started, this saves a lot of time [12]. The saved model can easily be transferred to the devices for inference. It can also be deployed in edge devices such as smartphones and IoT devices.

Alert Simulation

An MYSQL trigger is a setup on a mask table that will alter the count on the city table for the zip code of the image. This data helps to find statistics and to analyze the data for finding the most vulnerable locations and giving alerts. The MYSQL table structure is as follows.

— mask (id, image, zipcode)

— city (zipcode, phone_number, count)

Once the count is updated, the Python script will check if the count is greater than 500; and if it returns true, then the message is sent to the phone_number in the city table using the Sinch API credentials. The architecture for sending SMS is given in Fig. 6.

Results and discussion

We considered Precision, Recall, F1-score, and Accuracy as the metrics to evaluate the performance of

*Table. Se*quential order of layers in network with output shape, number of parameters used

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 76, 76, 64)	4864
batch_normalization_1	(None, 76, 76, 64)	256
conv2d_2 (Conv2D)	(None, 72, 72, 64)	102464
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
batch_normalization_2	(None, 36, 36, 64)	256
dropout_1 (Dropout)	(None, 36, 36, 64)	0
conv2d_3 (Conv2D)	(None, 32, 32, 128)	204928
batch_normalization_3	(None, 32, 32, 128)	512
conv2d_4 (Conv2D)	(None, 28, 28, 128)	409728
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0
batch_normalization_4	(None, 14, 14, 128)	512
dropout_2 (Dropout)	(None, 14, 14, 128)	0
conv2d_5 (Conv2D)	(None, 10, 10, 256)	819456
batch_normalization_5	(None, 10, 10, 256)	1024
conv2d_6 (Conv2D)	(None, 6, 6, 256)	1638656
max_pooling2d_3 (MaxPooling2D)	(None, 3, 3, 256)	0
batch_normalization_6	(None, 3, 3, 256)	1024
dropout_3 (Dropout)	(None, 3, 3, 256)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 256)	590080
batch_normalization_7	(None, 256)	1024
dropout_4 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 60)	15420
batch_normalization_8	(None, 60)	240
dropout_5 (Dropout)	(None, 60)	0
dense_3 (Dense)	(None, 1)	232

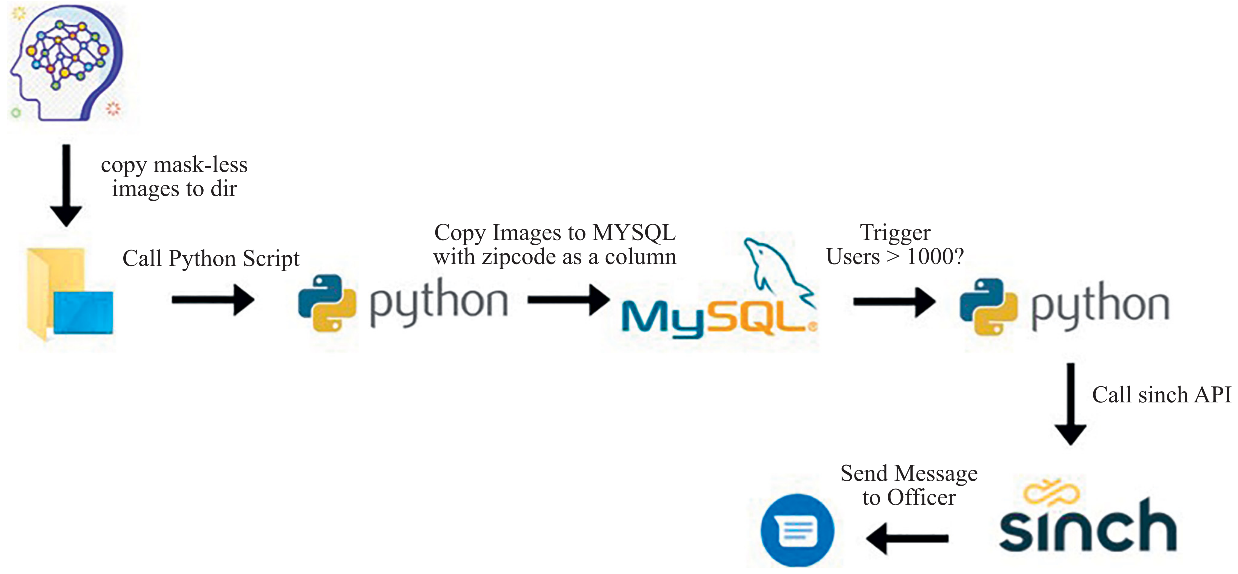

Fig. 6. Sending SMS if violations limit is reached in a location

the model in both the training and the testing phases. The values of metrics can be generated using these formulae as follows, where “TP” stands for True Positives; “FP” stands for False Positives; “TN” stands for True Negatives; and “FN” stands for False Negatives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{F1} - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The proposed mode is applied on Kaggle’s dataset and Prajnasb dataset. After training and obtaining the results on the test set, the best performing model is considered as the Teacher model after the Knowledge distillation [13–18]. The recall is observed 67 % using Max pooling and 57 % by Average Pooling for with-mask data whereas the recall is found to be same in both the cases for without-mask data. Precision is found to be the same in both cases for with-mask data; whereas the precision is 98 % by Max Pooling, and 97 % by using Average Pooling for without-mask data. F1-score is observed as 80 % using Max Pooling and 73 % using the Average Pooling for with-mask data; whereas it is 98 % for Max Pooling and 98 % for Average Pooling for without-mask data. Comparing both data sets, it is observed that Max Pooling is giving better results compared to Average Pooling by 1 % with Precision, 10 % with Recall. On average, Max Pooling is giving better F1-Score compared to the Average Pooling by 4 %. We also observed 1 % more accuracy using the Max Pooling. Hence, we found the Max Pooling is giving better results when compared to Average Pooling in all the metrics. Hence, the Student model is trained with the help of the Teacher model which in this case is the model built with Max Pooling as the pooling layer in the CNN architecture. Fig. 7, *a* shows the performance of pooling layer with mask label and Fig. 7, *b* shows the performance of pooling layer without mask label.

The inferencing will be continuously happening on the model as it is a video and the frames are constantly fed to the model for inference [19–24]. The model should predict the results faster as well as accurately as shown in Fig. 8. To maintain this tradeoff, the complexity of the model should not be too high and the accuracy of the model should also be taken care [25–28]. Knowledge distillation can be applied to our model to create a Student model which is smaller in size and it is also as accurate as of the Teacher model. After creating a Student model, the weights of the model are saved and can be used in different applications.

We have observed the increase in the accuracy from the Student model as the epochs are progressed, and finally the Student model gave 98 % accuracy after the Knowledge distillation for the Teacher model using the Max Pooling layer.

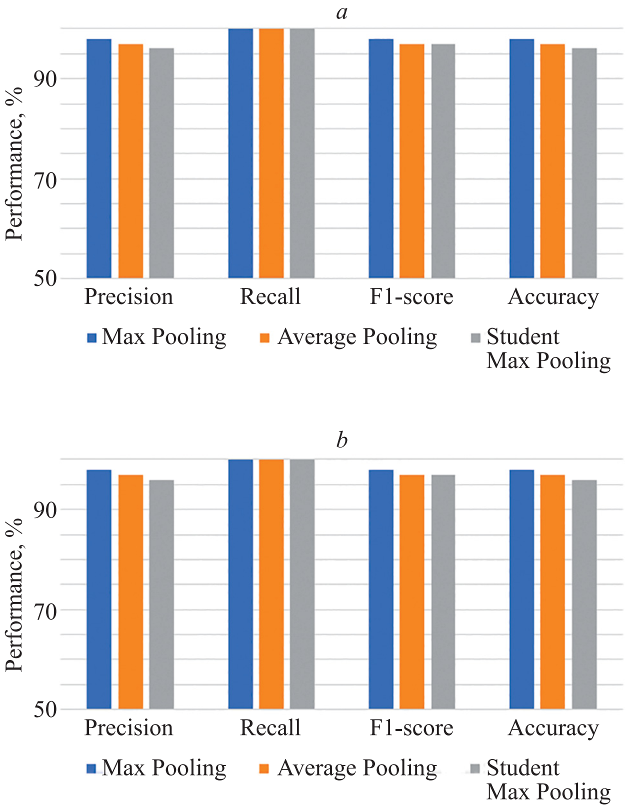

Fig. 7. Performance of merging layers for a label: with a mask (*a*); without a mask (*b*)

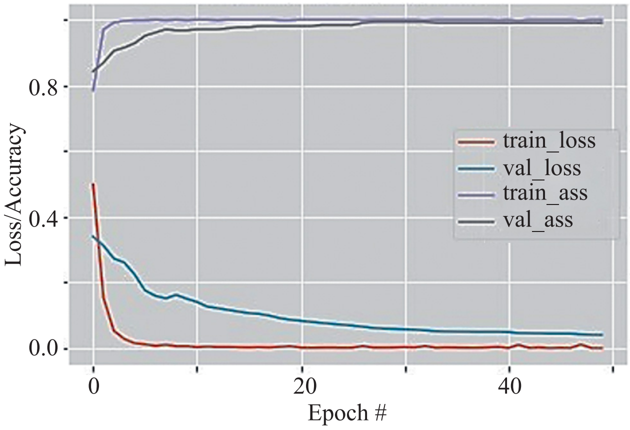

Fig. 8. Training and Validation loss for Student model

Conclusion

In summary, we propose a lightweight model for face mask detection using Knowledge distillation. The need to increase awareness about wearing masks is increasing rapidly, and to make the rules stricter, the government is imposing fines to people who are violating the rules for not wearing a mask [15]. This cloud-based approach makes the job a lot easier and provides timely alerts and notifications that help in identifying the danger zones and take necessary precautions by implementing lock-down in those areas and so forth [16]. The same architecture can be implemented to identify the people who are sick and coughing in the public places using the respective machine learning models and create a secure environment for the public [17]. The work achieves an accuracy of 98 % using Max Pooling layer in predicting the mask-less people, and there is a scope for improving the accuracy to implement the model in real time. In future, we plan to apply other detection methods such as R-CNN to detect the exact type of the mask.

References

1. Nagrath P., Jain R., Madan A., Arora R., Kataria P., Hemanth J. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society*, 2021, vol. 66, pp. 102692. https://doi.org/10.1016/j.scs.2020.102692
2. Matrajt L., Leung T. Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease. *Emerging Infectious Diseases*, 2020, vol. 26, no. 8, pp. 1740–1748. https://doi.org/10.3201/eid2608.201093
3. Fu C., Liu W., Ranga A., Tyagi A., Berg A. DSSD: deconvolutional single shot detector model. *arXiv*, 2017, arXiv:1701.06659. https://doi.org/10.48550/arXiv.1701.06659
4. Lin T.Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S. Future pyramid networks for object detection. *Proc. of the 30th IEEE Conference Proceedings on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944. https://doi.org/10.1109/CVPR.2017.106
5. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted residuals and linear bottlenecks. *Proc. of the 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
6. Farfade S.S., Saberian M.J., Li L. Multi-view face detection using deep convolutional neural networks. *Proc. of the 5th ACM International Conference on Multimedia Retrieval (ICMR)*, 2015, pp. 643–650. https://doi.org/10.1145/2671188.2749408
7. Chen S., Zhang C., Dong M., Le J., Rao M. Using ranking-CNN for age estimation. *Proc. of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 742–751. https://doi.org/10.1109/CVPR.2017.86
8. Hinton G., Vinyals O., Dean J. Distilling the knowledge in a neural network. *arXiv*, 2015, arXiv:1503.02531. https://doi.org/10.48550/arXiv.1503.02531
9. Bandaru A., Bhadani A., Sinha A. A facemask detector using machine learning and image processing techniques. *Engineering Science and Technology an International Journal*, 2020.
10. Gurav O. *Face Mask Detection Dataset. 2020*. Available at: https://www.kaggle.com/omkargurav/face-mask-dataset (accessed: 25.12.2021)
11. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, vol. 39, no. 6, pp. 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
12. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. V. 1, 2001, pp. I511–I518. https://doi.org/10.1109/CVPR.2001.990517
13. Oro D., Fernández C., Saeta J.R., Martorell X., Hernando J. Real-time GPU-based face detection in HD video sequences. *Proc. of the IEEE International Conference on Computer Vision Workshops (ICCV)*, 2011, pp. 530–537. https://doi.org/10.1109/ICCVW.2011.6130288
14. Glass R.J., Glass L.M., Beyeler W.E., Min H.J. Targeted social distancing designs for pandemic influenza. *Emerging Infectious Diseases*, 2006, vol. 12, no. 11, pp. 1671–1681. https://doi.org/10.3201/eid1211.060255
15. Masita K.L., Hasan A.N., Satyakama P. Pedestrian detection using R-CNN object detector. *IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2018, pp. 8625210. https://doi.org/10.1109/LA-CCI.2018.8625210
16. Girshick R. Fast R-CNN. *Proc. of the 15th IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448. https://doi.org/10.1109/ICCV.2015.169
17. Zhu X., Ramanan D. Face detection, pose estimation, and landmark localization in the wild. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2879–2886. https://doi.org/10.1109/CVPR.2012.6248014
18. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. MobileNets: Efficient

Литература

1. Nagrath P., Jain R., Madan A., Arora R., Kataria P., Hemanth J. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 // Sustainable Cities and Society. 2021. V. 66. P. 102692. https://doi.org/10.1016/j.scs.2020.102692
2. Matrajt L., Leung T. Evaluating the effectiveness of social distancing interventions to delay or flatten the epidemic curve of coronavirus disease // Emerging Infectious Diseases. 2020. V. 26. N 8. P. 1740–1748. https://doi.org/10.3201/eid2608.201093
3. Fu C., Liu W., Ranga A., Tyagi A., Berg A. DSSD: deconvolutional single shot detector model // arXiv. 2017. arXiv:1701.06659. https://doi.org/10.48550/arXiv.1701.06659
4. Lin T.Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S. Future pyramid networks for object detection // Proc. of the 30th IEEE Conference Proceedings on Computer Vision and Pattern Recognition (CVPR). 2017. P. 936–944. https://doi.org/10.1109/CVPR.2017.106
5. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted residuals and linear bottlenecks // Proc. of the 31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2018. P. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
6. Farfade S.S., Saberian M.J., Li L. Multi-view face detection using deep convolutional neural networks // Proc. of the 5th ACM International Conference on Multimedia Retrieval (ICMR). 2015. P. 643–650. https://doi.org/10.1145/2671188.2749408
7. Chen S., Zhang C., Dong M., Le J., Rao M. Using ranking-CNN for age estimation // Proc. of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P. 742–751. https://doi.org/10.1109/CVPR.2017.86
8. Hinton G., Vinyals O., Dean J. Distilling the knowledge in a neural network // arXiv. 2015. arXiv:1503.02531. https://doi.org/10.48550/arXiv.1503.02531
9. Bandaru A., Bhadani A., Sinha A. A facemask detector using machine learning and image processing techniques // Engineering Science and Technology an International Journal. 2020.
10. Gurav O. Face Mask Detection Dataset. 2020 [Электронный ресурс]. URL: https://www.kaggle.com/omkargurav/face-mask-dataset (дата обращения: 25.12.2021).
11. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. V. 39. N 6. P. 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
12. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). V. 1. 2001. P. I511–I518. https://doi.org/10.1109/CVPR.2001.990517
13. Oro D., Fernández C., Saeta J.R., Martorell X., Hernando J. Real-time GPU-based face detection in HD video sequences // Proc. of the IEEE International Conference on Computer Vision Workshops (ICCV). 2011. P. 530–537. https://doi.org/10.1109/ICCVW.2011.6130288
14. Glass R.J., Glass L.M., Beyeler W.E., Min H.J. Targeted social distancing designs for pandemic influenza // Emerging Infectious Diseases. 2006. V. 12. N 11. P. 1671–1681. https://doi.org/10.3201/eid1211.060255
15. Masita K.L., Hasan A.N., Satyakama P. Pedestrian detection using R-CNN object detector // IEEE Latin American Conference on Computational Intelligence (LA-CCI). 2018. P. 8625210. https://doi.org/10.1109/LA-CCI.2018.8625210
16. Girshick R. Fast R-CNN // Proc. of the 15th IEEE International Conference on Computer Vision (ICCV). 2015. P. 1440–1448. https://doi.org/10.1109/ICCV.2015.169
17. Zhu X., Ramanan D. Face detection, pose estimation, and landmark localization in the wild // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2012. P. 2879–2886. https://doi.org/10.1109/CVPR.2012.6248014
18. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. MobileNets: Efficient

convolutional neural networks for mobile vision applications. *arXiv*, arXiv:1704.04861, 2017. https://doi.org/10.48550/arXiv.1704.04861

19. Nagrath P., Jain R., Madan A., Arora R., Kataria P., Hemanth J. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society*, 2021, vol. 66, pp. 102692. https://doi.org/10.1016/j.scs.2020.102692
20. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. MIT Press, 2016, 800 p.
21. Jiang M., Fan X., Yan H. Retina Mask: A Face Mask detector. *arXiv*, 2020, arXiv:2005.03950v2. https://doi.org/10.48550/arXiv.2005.03950
22. Jiang X., Gao T., Zhu Z., Zhao Y. Real-Time Face Mask Detection Method Based on YOLOv3. *Electronics*, 2021, vol. 10, no. 7, pp. 837. https://doi.org/10.3390/electronics10070837
23. Liu S., Again S.S. COVID-19 face mask detection in a crowd using multi-model based on YOLOv3 and hand-crafted features. *Proceedings of SPIE*, 2021, vol. 11734, pp. 117340–117340. https://doi.org/10.1117/12.2586984
24. Sen S., Sawant K. Face mask detection for covid_19 pandemic using pytorch in deep learning. *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1070, pp. 012061. https://doi.org/10.1088/1757-899X/1070/1/012061
25. Sethi S., Kathuria M., Kaushik T. A real-time integrated face mask detector to curtail spread of coronavirus. *Computer Modeling in Engineering & Sciences*, 2021, vol. 127, no. 2, pp. 389–409. https://doi.org/1032604/cmes.2021.014478
26. Srivastava P., Khan R. A review paper on cloud computing. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2018, vol. 8, no. 6, pp. 17. https://doi.org/10.23956/ijarcsse.v8i6.711
27. Susanto S., Putra F.A., Analia R., Suciningtyas I.K.L.N. The face mask detection for preventing the spread of COVID-19 at Politeknik Negeri Batam. *Proc. of the 3rd International Conference on Applied Engineering (ICAE)*, 2020, pp. 9350556. https://doi.org/10.1109/ICAE50557.2020.9350556
28. Wang Z., Wang G., Huang B., Xiong Z., Hong Q., Wu H., Yi P., Jiang K., Wang N., Pei Y., Chen H., Miao Y., Huang Z., Liang J. Masked face recognition dataset and application. *arXiv*, 2020, arXiv:2003.09093v2. https://doi.org/10.48550/arXiv.2003.09093

convolutional neural networks for mobile vision applications // arXiv. arXiv:1704.04861. 2017. https://doi.org/10.48550/arXiv.1704.04861

19. Nagrath P., Jain R., Madan A., Arora R., Kataria P., Hemanth J. SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2 // Sustainable Cities and Society. 2021. V. 66. P. 102692. https://doi.org/10.1016/j.scs.2020.102692
20. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.
21. Jiang M., Fan X., Yan H. Retina Mask: A Face Mask detector // arXiv. 2020. arXiv:2005.03950v2. https://doi.org/10.48550/arXiv.2005.03950
22. Jiang X., Gao T., Zhu Z., Zhao Y. Real-Time Face Mask Detection Method Based on YOLOv3 // Electronics. 2021. V. 10. N 7. P. 837. https://doi.org/10.3390/electronics10070837
23. Liu S., Again S.S. COVID-19 face mask detection in a crowd using multi-model based on YOLOv3 and hand-crafted features // Proceedings of SPIE. 2021. V. 11734. P. 117340–117340. https://doi.org/10.1117/12.2586984
24. Sen S., Sawant K. Face mask detection for covid_19 pandemic using pytorch in deep learning // IOP Conference Series: Materials Science and Engineering. 2021. V. 1070. P. 012061. https://doi.org/10.1088/1757-899X/1070/1/012061
25. Sethi S., Kathuria M., Kaushik T. A real-time integrated face mask detector to curtail spread of coronavirus // Computer Modeling in Engineering & Sciences. 2021. V. 127. N. 2. P. 389–409. https://doi.org/1032604/cmes.2021.014478
26. Srivastava P., Khan R. A review paper on cloud computing // International Journal of Advanced Research in Computer Science and Software Engineering. 2018. V. 8. N 6. P. 17. https://doi.org/10.23956/ijarcsse.v8i6.711
27. Susanto S., Putra F.A., Analia R., Suciningtyas I.K.L.N. The face mask detection for preventing the spread of COVID-19 at Politeknik Negeri Batam // Proc. of the 3rd International Conference on Applied Engineering (ICAE). 2020. P. 9350556. https://doi.org/10.1109/ICAE50557.2020.9350556
28. Wang Z., Wang G., Huang B., Xiong Z., Hong Q., Wu H., Yi P., Jiang K., Wang N., Pei Y., Chen H., Miao Y., Huang Z., Liang J. Masked face recognition dataset and application // arXiv. 2020. arXiv:2003.09093v2. https://doi.org/10.48550/arXiv.2003.09093

Authors

Vattumilli Komal Venugopal — B.Tech., Student, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, India, https://orcid.org/0000-0002-9292-7473, komalvenugopal@gmail.com

Movva Lalith — B.Tech., Student, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, India, https://orcid.org/0000-0002-1767-3186, lalithindian@gmail.com

Thangavelu Arun Kumar — Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, India, SC 57055634400, https://orcid.org/0000-0001-7384-8975, arunkumart@vit.ac.in

Jayaraman Jayashree — Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, India, SC 57191669840, https://orcid.org/0000-0002-0191-8070, jayashree.j@vit.ac.in

Jayaraman Vijayashree — Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, India, SC 57190985580, https://orcid.org/0000-0001-6987-3377, vijayashree.j@vit.ac.in

Авторы

Комал Венугопал Ваттумилли — студент, Школа компьютерных наук и инженерии, Технологический институт Веллору, Веллуру, 632014, Индия, https://orcid.org/0000-0002-9292-7473, komalvenugopal@gmail.com

Лалит Мовва — студент, Школа компьютерных наук и инженерии, Технологический институт Веллору, Веллуру, 632014, Индия, https://orcid.org/0000-0002-1767-3186, lalithindian@gmail.com

Арун Кумар Тангавелу — профессор, Школа компьютерных наук и инженерии, Технологический институт Веллору, Веллуру, 632014, Индия, SC 57055634400, https://orcid.org/0000-0001-7384-8975, arunkumart@vit.ac.in

Джаяшри Джаяраман — профессор, Школа компьютерных наук и инженерии, Технологический институт Веллору, Веллуру, 632014, Индия, SC 57191669840, https://orcid.org/0000-0002-0191-8070, jayashree.j@vit.ac.in

Виджаяшри Джаяраман — профессор, Школа компьютерных наук и инженерии, Технологический институт Веллору, Веллуру, 632014, Индия, SC 57190985580, https://orcid.org/0000-0001-6987-3377, vijayashree.j@vit.ac.in

Received 20.02.2022
Approved after reviewing 25.03.2022
Accepted 19.05.2022

Статья поступила в редакцию 20.02.2022
Одобрена после рецензирования 25.03.2022
Принята к печати 19.05.2022

Работа доступна по лицензии
Creative Commons
«Attribution-NonCommercial»