A face detection ensemble to monitor the adoption of face masks inside the public transportation during the COVID-19 pandemic

João Paulo Canário1 · Marcos Vinícius Ferreira2 · Junot Freire2 · Matheus Carvalho3 · Ricardo Rios1

Received: 14 December 2020 / Revised: 22 February 2022 / Accepted: 9 March 2022 / Published online: 20 April 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
The designing of ensembles is widely adopted when single machine learning methods fail to obtain satisfactory performances by analyzing complex data characterized by being imbalanced, high-dimensional, and noisy. Such a failure is a well-known statistical challenge when the learning algorithm searches for a model in a large space of hypotheses and the data do not significantly represent the problem, thus not inducing it from a space of admissible functions towards the best global model. We have addressed this issue in a real-world application, whose main objective was to identify whether users were wearing masks inside public transportation during the COVID-19 pandemic. Several studies have already pointed that face masks are an important and efficient non-pharmacological strategy to reduce the virus spread. In this sense, we designed an approach using Convolutional Neural Networks (CNN) to track the adoption of masks in different transportation lines, regions, days, and time. Aiming at reaching this goal, we propose an ensemble of face detectors and a CNN architecture, called MaskNet, to analyze all public-transport passengers and provide valuable information to policymakers, which are able to dedicate efforts to more effective advertisements and awareness work. In practice, our approach is running in a real scenario in Salvador (Brazil).

Keywords Face detection · Mask detection · Ensemble models · Covid-19 · Deep Learning

1 Introduction
The coronavirus disease 2019 (COVID-19) is a severe acute respiratory syndrome caused by the coronavirus 2 (SARS-CoV-2), which has deeply affected not only the health systems but also the global economy and politics. Due to its rapid spread, the World Health Organization (WHO) has declared a pandemic outbreak on March 11th, 2020 [7]. According to the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE), up to
November 3rd, 2020, the number of confirmed cases around the world has surpassed more than 47 million cases and the number of deaths is greater than 1.2 million registers.

Aiming at dealing with such a very aggressive virus, scientists have been dedicating efforts in two main directions. The first one is the development of a new vaccine specifically designed to immunize the population. Based on the website Coronavirus Vaccine Tracker published by The New York Times\(^1\), on November 3rd, 2020, there exists 6 vaccines already approved for early or limited use, 11 ones in Phase 3, being assessed in large-scale efficacy tests, and 50 others in the initial stages (Phases 1 and 2). The second direction is related to non-pharmacological strategies, which include, for example, social distancing, quarantine, isolation, and the adoption of alcohol-based hand sanitizers and face masks \([7, 9]\).

Researches focused on non-pharmacological strategies are specially necessary to keep the population safer while vaccines are not concluded or being widely applied. Moreover, all conclusions and learning on this topic will be also necessary to reduce the impact of new infection waves and further health problems. For instance, Markel et al. \([30]\) have risen a critical question about the role played by non-pharmaceutical interventions in delaying the overall and peak attack rate, and reducing the number of cumulative deaths during the 1918-1919 Influenza Pandemic in US cities. According to the authors, their findings demonstrated a strong relationship between early, sustained, and layered application of non-pharmaceutical interventions and the mitigation of the influenza pandemic consequences. Their scientific conclusion emphasizes non-pharmaceutical interventions should be considered in planning for future severe influenza pandemics.

Among all possible non-pharmacological strategies, the adoption of face masks has been recommended by the WHO and scientists, that have been conducting researches to understand their influence to reduce the person-to-person spread through close contact. In this sense, Chu et al. \([8]\) identified 172 observational studies, on 25,697 patients, across 16 countries, and six continents, which suggest the adoption of face masks protects both health-care workers and the general public against infection by coronavirus. Complementary, a scientific report published in \([41]\) details several studies that show important relative and absolute benefits of wearing face masks. Still according to the researchers, policymakers might recommend the adoption of masks, mainly in densely populated areas that have high infection rates as, for instance, USA, India, Brazil, and South Africa, once their usage will probably outweigh any potential downsides.

In Brazil, where the high number of confirmed cases and deaths caused by coronavirus has demanded to a social isolation for a period greater than 8 months\(^2\), the public transportation, widely used by the population in general, is an important monitoring point. The relaxation of the mask adoption in such an environment can be responsible for increasing the reproduction number (R) of SARS-CoV-2. For example, before starting the pandemic outbreak, more than 1,1 million passengers used to take buses per day in Salvador – Brazil (average calculated between 2016 and 2019). In this context, the restriction of minimum distance among passengers cannot be assured, especially during rush hours. Additionally, the monitoring of people wearing face masks by human supervisors is neither efficient, due to the huge volume of passengers, nor economically possible.

Therefore, the implementation of an Artificial Intelligence (AI) system, devoted to identify whether or not every passenger is respecting the mask recommendation, is essential for, at least, two main reasons: (i) to support the development of more effective policies, once

---

\(^1\)https://www.nytimes.com/interactive/2020/science/coronavirus-vaccine-tracker.html

\(^2\)By considering the local policies on November 03rd, 2020.
the main awareness campaigns can be dedicated to regions where the rate of the facemask-wearing adoption is getting lower; and (ii) to present lines “safer” to users where face masks have been more adopted.

In this work, our main goal was the development of an AI solution based on Deep Neural Network (DNN) and Visualization to monitor all bus passengers in Salvador (Brazil). Aiming at reaching this goal, we have created a new DNN architecture to detect whether a passenger wears a face mask. Due to the importance of this topic, several researchers are focused on building up approaches, usually designed on top of DNNs, to verify, for example, the correct usage of masks [38], the adoption of masks in specific places [34], and the mask monitoring in places with the high flow of people. In all approaches, the first step is the face detection [25, 43], which is responsible for identifying a portion of an image referred to as a region of interest (ROI). Finally, after knowing faces in images, DNNs are also applied to classify the presence/absence of masks [24].

Although there exists several approaches developed to detect faces, in our experiments, we have noticed the performance of the main ones was limited to an accuracy of 80%. Indeed, this result is very relevant due to the dynamic and complex set of images collected from buses, usually characterized by different illumination, wrongly positioned cameras, unfocused and rotated photos, and faces partially covered by glasses, hats, another person, so on.

Aiming at solving this situation and meeting our goal, we have designed two main contributions. The first one is an ensemble that combines different detectors to maximize the chances of finding faces in images. Our second contribution, after detecting faces, is the analysis of a high data flow to predict the mask usage. Once our approach performs this prediction with high accuracy, we were able to create a data visualization system that summarizes the extracted information and combines them along with geospatial and temporal details. According to our results, we emphasize that our approach to detect faces provides a relevant contribution to the state-of-the-art in the Computer Vision area. Moreover, by using our visual analytics system, governments can create more effective policies that propose the adoption of specific non-pharmacological strategies based on the population behavior in different regions, days of the week, and times.

This manuscript is organized as follows: Section 2 describes the main related work; in Section 3, we show our approaches to detect face and classify the according to the mask usage; Sections 4 and 5 detail the experimental setup and the obtained results; finally, in Section 6, we discuss our results and draw the final conclusions.

2 Related work

Aiming at ratifying the importance of this work to deal with COVID-19, in this section, we show a set of studies recently published to detect faces and the adoption of masks. According to the survey published by Zafeiriou et al. [52], face detection is widely studied in computer vision literature, due to its challenging nature and the countless applications that require it as a first step. Based on the taxonomy presented by the authors, Face Detection algorithms are essentially designed on top of rigid templates and deformable parts-model. The first one learns from boosting-based methods as, for instance, the application of Deep Neural Networks. The latter uses deformable models to describe faces by their main parts. In our work, we have considered algorithms, which implement rigid templates as usually considered by the current state of the art.
One of the most well-known methods designed to detect face was proposed by Yang et al. [51], referred to as MTCNN (Multi-task Cascaded Convolutional Networks), that uses two Convolution Neural Network (CNN) architectures. The first one produces candidate marks, in which the face might be represented. The next CNN works as a filter, thus refining the marks to better identify the face. In summary, CNN behaves as an ordinary Artificial Neural Network (ANN), whose the main difference is the great number of neurons and hidden layers that enable it to recognize patterns with extreme variability, distortions, and geometric transformations [5, 27] (more details are provided in Section 3.3).

Luo et al. [29] also proposed a CNN architecture to detect face. Their main contribution is a cascade model with three stages of DNNs that uses an iterative bounding-box regression to improve face marks in images. The authors consider an inherent correlation of classification and bounding-box regression to reduce the face detection error.

Nieto-Rodríguez et al. [34] designed a study similar to ours aiming at detecting when the face mask is not worn in the operating room. Instead of using CNN, their approach is based on a previous work published by Viola and Jones [46], who proposed a robust method to detect faces in real time. Essentially, the method uses HSV filters to preprocessing the images by taking into account different values for hue, saturation, brightness, which are later used to train an AdaBoost-based model called LogitBoost. According to the manuscript, the classification results reached an accuracy of 95%.

The SARS-CoV-2 outbreak has motivated the adoption of such researches to support non-pharmacological strategies. For example, Kolhar et al. [23] developed a biometric solution that uses CNN and IoT (Internet of Things) devices to recognize people wearing masks. The authors use MTCNN to detect faces, which is later taken into account to extract reference points. Those points are, then, used to find out similar faces previously stored in a database. The classification accuracy presented by the authors is around 82%. Din et al. [44] have also published a work to recognize faces even using masks. In their work, the face region covered by the mask is reconstructed by a U-Net architecture. To reach this goal, a first model is used to segment the mask region and the next one is adopted to reconstruct it.

Another important work was published by Loey et al. [28], in which a Deep Neural Network (DNN), referred to as ResNet50, was created to analyze images and extract face characteristics. Next, the characteristics were used to train different supervised Machine Learning (ML) models as, for instance, Support Vector Machine (SVM) and Decision Tree (DT). By using SVM, the authors have got an accuracy greater than 99.6%.

Qin and Li [38] have proposed to identify masks in super-resolution images. The authors recommend a preprocessing step to reduce the variance and contrast and slightly increase the saturation in 1%. MTCNN is, then, used to detect faces in every processed image. Finally, a DNN, called Mobilenet-v2, was designed to analyze all faces and detect the presence/absence of masks. According to their results, the classification accuracy was greater than 98%.

In our scenario, we have faced an important restriction to identify masks. The images automatically taken in the buses have low resolutions and are affected by different environmental conditions as, for instance, illumination, focus, and occlusions. The adoption of traditional models to identify faces in our images has presented results noticeably lower than the state of the art summarized in this section. This issue has motivated us to create an ensemble that, instead of using a single method, takes advantage of the combination of different face detectors.
3 Mask detection with face detection ensemble

In this work, we deployed our full mask detection approach by using a REST API service to asynchronously process batches of images and, consequently, scale our service in multiple machines. Aiming at reaching this goal, we have considered the following stack of technologies: 1) Docker - an open platform for developing, shipping, and running applications, which allows splitting the applications in different infrastructures and performing a fast software deployment [32]; 2) Celery - a task queue library for Python web applications used to asynchronously execute work outside the HTTP request-response cycle [6]; 3) FastAPI - a modern, fast, high-performance, web framework for building APIs in Python [11]; 4) OpenCV - an open source computer vision and machine learning software library [2]; 5) Scikit-Learn - a open source machine learning library [37]; and 6) TensorFlow - a free and open-source library for machine learning [1]. In our system design, the celery workers are responsible for sending asynchronous batches of images in parallel to the REST API service, which runs the mask detection pipeline as depicted in Fig. 1.

Firstly, before sending a set of images to the pipeline (Fig. 1), we preprocess them to improve their quality and, then, increase the models performance. Next, the first step of the pipeline is responsible for filtering those images with poor conditions. The second pipeline step contains the main contribution of our proposal, which is responsible for detecting faces. In our experiments, we have analyzed a set of face detectors mostly used in the literature. Based on our results, we have noticed the accuracy could be improved when detectors are combined instead of using just a single one. To proceed with this contribution, we have built up an ensemble [10] that unifies regions of interest (ROI) identified by different face detectors. In the final pipeline step, we have created a CNN architecture to classify whether public transportation users are wearing face masks. The following sections provide more details about the complete pipeline.

![Fig. 1 Illustration of the system designed to detect faces and masks. The Celery workers send image batches to be processed by the REST API that will detect the usage of masks in public transportation](image-url)
3.1 Image preprocessing

In this phase, we used a set of techniques widely considered in Computer Vision problems. In summary, such techniques clean and transform the data input to maximize the learning process. In our scenario, pictures taken inside the buses are affected by cameras with low quality, current weather and road condition, natural movements, and different illumination depending on the time of the day, thus affecting the image conditions and producing, for example, underexposed, overexposed, and blurry images.

Aiming at improving the image quality, we have considered the Contrast Limited Adaptive Histogram Equalization (CLAHE) \[54\], which is widely adopted in image preprocessing. This method starts equalizing the color histogram from small image blocks, which are later used to calculate the Cumulative Probability Distribution (CPD). Then, CPD is used to convert gray levels into a uniform distribution function. These steps allow changing the image contrast in different scales. However, the contrast is limited by the influence of noises. To reduce such influences, CLAHE defines a histogram threshold as a clip limit, which uniformly redistributes all pixels greater than it to their neighborhood. In summary, this method transforms images by changing the contrast in different scales, redistributing the bright, and reducing the noise. Its adoption is reported in the literature to improve images take from, for example, medicine \[17, 20\], underwater \[18\], and face recognition \[13\]. In our images, the best configuration suggested by our empirical analyses, after widely varying the parameters, was image blocks with a size of \(8 \times 8\) and clip limit equals to \(40\).

Even improving the images with the CLAHE method, we have noticed that, in images overexposed or underexposed, the face and mask detectors still present a poor performance. Therefore, as a project definition, we decided to remove all images in such conditions. Their detection, though, was performed by considering two strategies based on filtering and Computer Vision (CV) discussed next.

3.2 First pipeline step: filtering

The initial strategy considered to detect images with poor quality has used a straightforward method based on histograms with bins representing a color scale from 0 to 255, in which the lower the value is, the darker is the image. After an extensive empirical evaluation in our images, we have fixed three intervals over the bins: \(i_1 = [0, 64], i_2 = [65, 210], i_3 = [211, 255]\). Next, we calculate the grayscales and check the interval derived from them. The experiments with our images emphasized the best ones are placed in \(i_2\), whereas \(i_1\) and \(i_3\) represent very dark and light, respectively. Therefore, we only detect faces and find masks in \(i_2\) images.

The second strategy uses image descriptors to train Machine Learning classifiers. Initially, we have used Local Binary Pattern (LBP) that is a visual descriptor based on a Texture Spectrum model \[16, 35\]. LBP starts calculating local textures by considering a base pixel \(I(x, y)\) and a radius \(R\). Hence, LBP binarizes \(N\) pixels \(R\)-distant from \(I(x, y)\) by changing all values to 0 or 1, if they are lower or greater than the base pixel, respectively. Next, the binary values \((b)\) are used to calculate a decimal number by considering, for example, a clock-wise approach to create a binomial factor as shown in (1).

\[
LBP = \sum_{p=0}^{N-1} b 2^p
\]
This method can be computationally expensive due to the values of \( R \) and \( N \). As a more efficient alternative, we have considered a convolutional filter designed to work as edge and emboss maps. In summary, the resultant texture map is obtained after multiplying regions of an image by a kernel (smaller matrix). For the edge and emboss filters, we used a 3 \( \times \) 3 kernel defined as \([0, 1, 0], [1, -4, 1], [0, 1, 0]\) and \([-2, -1, 0], [-1, 1, 1], [0, 1, 2]\), respectively. Figure 2(A) illustrates an image from our private dataset transformed in gray scale and its histogram is shown in Fig. 2E. Figures (B – D) show its transformations using LBP, and the edge and emboss convolutional kernels, respectively, along with their histograms in Figures (F – H). In all plots, the histograms were normalized to [0, 1].

Finally, we use the histograms to train the following classifiers: Support Vector Machine (SVM); Multilayer Perceptron (MLP); and Random Forest (RF). SVM is a classifier built up on top of the Statistical Learning Theory (SLT) [45], which provides a framework to create hyperplanes to separate instances in different classes. In our experiments, the SVM model was trained using a radial kernel and the regularization parameter equals to 2.0.

MLP is a well-known type of artificial neural network composed of, at least, three layers: input, hidden, and output. The neurons adopted in such network use activation functions, whose outputs are connected to neurons from the following layer to create a nonlinear mapping [14]. In our scenario, we have created 375 neurons in the hidden layer with ReLu (better explained later in this manuscript), as activation function, and Adam optimization, learning rate equals to 0.001, and 200 training epochs.

Finally, our last model was based on RF, which is an ensemble that combines a set of decision trees to classify unseen examples based on a voting strategy [4]. The term ensemble is used to refer to the aggregation of several individual classifiers in order to provide the final prediction. The RF model was trained using a total of 100 trees, a minimum size of leaves equals to 3, and the number of variables randomly sampled as candidates at each split as 2. It is worth emphasizing that the parameters for all models were set after performing an empirical analysis over different configurations.

Fig. 2 (A) An image in gray scale from our real-world dataset, and its transformation using LBP, and the edge and emboss convolutional kernels (B – D), respectively. Plots (E – H) shows their histograms.
3.3 Second pipeline step: face detection

The second pipeline step uses Artificial Neural Networks to perform two main tasks: (i) classification – to estimate the probability of finding a face in the image; and (ii) localization – to define the coordinates of the bounding box in which the face is located in the image.

As discussed by Yang et al. (2020) [51], Machine Learning has been a prominent area to improve researches in Computer Vision. Among all available models, Artificial Neural Network (ANN), specifically Deep Neural Network (DNN), has been widely adopted to the task of object detection [48, 49]. In summary, ANN is a bio-inspired approach that uses mathematical functions to simulate natural neurons and their activation systems [14]. Normally, neurons are connected to form different layers (e.g. input, hidden, and output), as shown in Fig. 3A, which are responsible for processing the input data, extract implicit information, and take some decision (e.g. perform some prediction). The learning is based on a generalization process that models weights among neurons as a new example is provided to the network [14].

The main difference between DNN and SNN (Shallow Neural Network) is the number of neurons and hidden layers, thus making the network architecture more complex and, as consequence, able to learn more details about the input data.

In our work, we focused our analysis on a specific DNN architecture referred to as Convolution Neural Network (CNN) [27], which produces great learning results from images. As shown in Fig. 3(B), CNNs are characterized by including new blocks in the hidden layer [5, 27] as, for instance: (i) convolutional – the most important block to perform the learning process, which is based on a set of learnable filters, typically designed to deal with information in 3 dimensions: width, height, and depth. The convolutional block produces activation maps that allow recognizing features as, for instance, edges, colors, and patterns; (ii) activation – set of functions (e.g. Rectified Linear Unit – ReLU) developed to avoid learning problems as, for instance, getting stuck near zero or indefinitely growing up; (iii) pooling – block created to perform downsampling operations in different spatial dimensions; and, finally, (iv) fully connected (dense) layers – blocks designed to compute the final class scores.

The face detection performed in our approach is based on four pre-trained CNN models: You Only Look Once (YOLO) [39], Multi-Task Cascaded Convolutional Neural Network (MTCNN) [53], ResNet [15], and Faster R-CNN [40].
The YOLO architecture is composed of 26 convolutional, 4 pooling, and 2 full-connected layers. The face identification is performed as a fast regression task, that provides great precision even analyzing every image once. In summary, YOLO splits the image as a grid, in which all cells are analyzed to predict a bounding box and a confidence level to estimate the presence of a sought object. The pre-trained model adopted in our analysis was based on the Wider Face dataset [50].

MTCNN is based on three steps with different CNN architectures. The first one, called P-Net, is composed of 3 convolutional (with 10, 16, and 32 filters) and 1 pooling layers. The second architecture is referred to as R-Net and contains 3 convolutional (with 28, 48, and 64 filters), 2 pooling, and 1 full-connected layers. The final one, known as O-Net, has 4 convolutional (with 32, 64, 64, and 128 filters), 3 pooling, and 1 full-connected layers. P-Net was designed to detect candidate faces, whose results are refined by R-Net, thus removing regions with a greater probability of having no face. Finally, O-Net provides a final face mark and 5 important reference points. In our experiments, we considered a pre-trained model adjusted on the Face Detection dataset and Benchmark [19], and the WIDER FACE dataset [50].

The third CNN model considered in our work was Faster ResNet, which is based on an implementation of the Dlib framework [21], specifically designed to detect and recognize faces. This model is totally based on ResNet (with 33 convolutional and full-connected layers), but the number of convolutional layers that was reduced to 26. The main advantage of this architecture is the residual learning that significantly reduces the training process (weight adjustment) [15]. For this network, we have trained it with about 3 million faces extracted from the scrub dataset [33], VGG dataset [36], and images published by researchers who proposed the Dlib framework.

The final model that we have assessed in our work was R-CNN, which is based on the Region Proposal Network (RPN) to predict bounding boxes and object scores. Aiming at reducing the computational costs, the model is based on shared memory, in which the feature map is available to all convolutional layers. The architecture proposed contains a fast ZF network [31], composed of 5 convolutional and 3 fully-connected layers, along with VGG-16 [42], which contains 13 convolutional and 3 full-connected layers. The R-CNN model was also pre-trained over the Wider Face dataset.

After performing scrutiny on those face detectors, we have noticed their performances were affected by the characteristics of our environment, since the passengers’ images are taken inside the buses in movement, under different illumination conditions, and using low-quality devices. Besides the performance variation, we also realized there was a strong divergence among the bounding boxes estimated by every face detector.

Such observations have called our attention and motivated us to create a face detector ensemble devoted to combine bounding boxes from different models. The pseudo-code of the proposed ensemble is shown in Algorithm 1, which receives as input an image and a set of face detector models. As output, it returns the boundaries of bounding boxes to represent all existing faces. Lines 2–4 estimate all bounding boxes from every individual face detector. Next, in line 6, we run the method Group Rectangles available in OpenCV [3], which can be used to cluster similar bounding boxes. This method executes a clustering algorithm that analyzes similarities among the locations and sizes of all bounding boxes. Such similarities are calculated by using a relative difference defined by the parameter \( \epsilon \). Due to the characteristic of our images, we have noticed this parameter could be set as 0.9. Moreover, we also set the threshold parameter \( \tau = 1 \) that is used by the clustering algorithm to return at least one group (bounding box in our case).
In line 6, let $C$ be the clustering output containing the boundaries from all similar bounding boxes, i.e., $C = \{[x_1, y_1, w_1, h_1], \ldots, [x_n, y_n, w_n, h_n]\}$ such that $|C| = n$. After knowing which bounding boxes were clustered, we use them in our ensemble by taking the expected value of their boundaries as shown in lines from 8 to 11. In summary, $x$ is the starting point on the axis $x$, $y$ is the starting point on the axis $y$, $w$ is the face width, and $h$ is the face height.

Algorithm 1 Ensemble algorithm to improve the face detection layer.

**Data:** $I$ = an image, $F = \{f_1, f_2, \ldots\}$ = a set of face detectors, $\epsilon$ = the relative difference to merge bounding boxes, $\tau$ = the minimum number of found face

**Result:** $[x_b, y_b, w_b, h_b]$ = bounding box boundaries estimated by the ensemble

1 /* Create a vector to store all bounding boxes yielded by the face detectors */
2 for $i$ in 1:$|F|$ do
3     $v[i] = f_i(I)$
4 end
5 /* Cluster the more similar bounding boxes */
6 $C = group(v, \epsilon, \tau)$
7 /* Compute the ensemble bounding box boundaries */
8 $x_b = E[X] = \sum_{x_i \in C} x_i P(X = x_i)$
9 $y_b = E[Y] = \sum_{y_i \in C} y_i P(Y = y_i)$
10 $w_b = E[W] = \sum_{w_i \in C} w_i P(W = w_i)$
11 $h_b = E[H] = \sum_{h_i \in C} h_i P(H = h_i)$
12 return $[x_b, y_b, w_b, h_b]$

3.4 Third pipeline step: mask detection

The last pipeline step presented in Fig. 1 also contains a CNN architecture, which was designed to predict whether the detected faces were wearing masks. The activation function considered in our hidden layers was ReLU and Sigmoid in the output one. ReLU is an activation function that sets all negative values to zero, while other values are simply kept. The Sigmoid function is presented in (2), in which $a$ is the slope parameter, $v$ is a neuron output usually defined as $v = \sum_{i} w_i \cdot x_i + b$ such that $w_i$ are synaptic weights to quantify the importance of every input $x_i$ and $b$ is an external bias to control the linear intersection point with the zero-crossing axis.

$$y = \frac{1}{1 + exp(-av)} \quad (2)$$

During the training, we have also considered the cross-entropy loss to quantify the classification errors, and the generalization (learning) process was conducted by Adam Optimizer [22] to update the weights and the learning rates. Finally, all strides present in the convolutional layers were set as $1 \times 1$ and our full architecture, referred to as MaskNet in this work, is shown in Fig. 4.

The first layer is a convolutional block with 32 filters and kernel size equals to $3 \times 3$. The following 4 convolutional blocks have a similar configuration, but the number of filters defined as 64. In between two convolutional blocks, we have included MAX pooling blocks with size equals to $2 \times 2$, which operate by taking the maximum value in every piece of the feature maps. After the feature extraction process, our architecture performs
the classification by combining a full-connected layer with 64 processing nodes, a dropout regularization with the learning rate set as 0.5, and another fully-connected layer with 2 nodes. The dropout regularization was important to reduce the possibility of overfitting by limiting the network weights, once higher values can lead to a more complex model.

We have trained MaskNet with 2 well-known scientific datasets plus a private one containing images automatically taken inside the public transportation in Salvador (Brazil). The first considered dataset, called Real-World Masked Face Dataset (RMFD)\(^3\), was created by researchers from Wuhan (China), who collected about 5,000 faces wearing masks and 90,000 with no masks. The second dataset\(^4\), referred in this paper to as Artificial Face Masks (AFM), contains 690 faces along with 686 faces with masks artificially inserted. The private database was collected by Integra, a company that manages public transportation in Salvador (Brazil), where this project has been conducted. The images were collected before and after April 23rd, 2020, when a local decree required the adoption of masks while using public transportation. It is worth emphasizing that a sample randomly extracted from those datasets were enough to successfully train our network: (i) RMFD – 2,156 (no mask) and 2,202 (with mask); (ii) AFM – 686 (no mask) and 689 (with mask); and (iii) Integra – 2,171 (no mask) and 3,004 (with mask). Hence, the final dataset was composed of 10,908 images, in which 5,013 and 5,895 represent without and with masks, respectively. The reader may notice that the sample from AFM is lower, once they are artificial and were included to intentionally add some disturbance to the actual faces wearing masks. Finally, we highlight all images were resized to 128 × 128, before being analyzed by our CNN. We have achieved great results with our network after 25 training epochs, 128 batch size, and a validation rate of 10%.

### 3.5 Evaluation metrics

The validation of our approach was performed by using three metrics: (i) accuracy; (ii) ROC (Receiver Operating Characteristic) curve; and IOU (Intersection Over the Union). In summary, accuracy computes the total number of correct classifications, represented by the sum of TP (true positive) and TN (true negative), divided by the total number of images, thus producing an overall performance result to compare distinct approaches.

\[
\text{Accuracy} = \frac{TP + TN}{n} \quad (3)
\]

\(^3\)https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset

\(^4\)https://github.com/prajnasb/observations
The ROC curve [12] plots TPR (True Positive Rate or Recall), (4) and FPR (False Positive Rate), (5). The closer to the point (0, 1) the curve is, i.e., as the area under the curve approaches to 1, the better is the final classification. The diagonal line in this plot simulates a random classifier, such that any result below it is worse than choosing any random class. In addition, we show the classification results as a confusion matrix, which presents TP, TN, FP (false positive), and FN (false negative) as well.

\[
TPR = \frac{TP}{TP + FN}\quad (4)
\]

\[
FPR = \frac{FP}{FP + TN}\quad (5)
\]

The IOU metric, presented in (6), was designed to assess images by comparing the output classifier and the expected result provided by specialists. In this equation, A is the ground truth, i.e., the bounding box defined by specialists, whereas B is the predicted bounding box. A perfect match between A and B yields a value equals to 1.

\[
IOU = \frac{\text{Intersection Area}}{\text{Union Area}} = \frac{A \cap B}{A \cup B}\quad (6)
\]

4 Experimental setup

Our setup was created by considering three experimental evaluation. Firstly, we evaluate the filtering pipeline step (Section 3.1) created to analyze images and remove those affected by different illumination conditions, which may jeopardize our face detection ensemble.

Next, we assess our main proposal that is the face detection ensemble (Section 3.3), designed to combine the different contributions of well-known face detectors to increase the chance of finding faces in our scenario. As a piece of remainder, we took advantage of a complex and dynamical scenario to create a new approach that, besides solving a practical issue, contributes to the state of the art in Computer Vision. All images considered in this part of our experimental setup were individually analyzed by human specialists who created bounding boxes later used as ground truth.

Finally, the last analyses were conducted to evaluate our CNN architecture (Section 3.4) to detect whether users were wearing masks in public transportation in Salvador (Brazil). Then, we also created visual metaphors that are useful to better understand and map the adoption of masks in different regions of Salvador.

The full implementation of the REST API service (see Fig. 1) proposed in our approach was performed on a server with a GPU TITAN V with 12GB, an Intel Core i9-9900K processor with 16 threads and a base frequency of 5GHz, and a memory size of 32GB. Table 1 summarizes the execution times collected from different scenario in order to better estimate the performance of our system. For every row in this table, we have repeated the experiment 35 times and calculated the mean values for a single image and all image batches. Therefore, by using our parallel system, we are able to significantly improve the speedup, i.e., the throughput to process all collected images.

The COVID-19 outbreak has demanded a quarantine in Salvador, thus significantly reducing the number of public transportation users. Figure 5 shows the number of pictures taken inside the buses per day, which are registered when the passengers cross the turnstile. Every event triggered by turnstiles produces four pictures that we use to find users’ faces, i.e., if faces are found in a picture, then we do not need to analyze the remaining ones. Therefore, by considering the volume of users and the processing capacity of our approach,
### Table 1  Temporal performance to detect faces and masks

| Batch size | Processing time (one image) | Mean processing time (batch) |
|------------|-----------------------------|------------------------------|
| 100        | 0.5710s                     | 00h00m57.1026s               |
| 200        | 0.3120s                     | 00h01m2.4038s                |
| 300        | 0.2276s                     | 00h01m8.2857s                |
| 400        | 0.1686s                     | 00h01m7.4427s                |
| 500        | 0.1901s                     | 00h01m35.0718s               |
| 600        | 0.2061s                     | 00h02m3.6743s                |
| 700        | 0.1759s                     | 00h02m3.1237s                |
| 800        | 0.1673s                     | 00h02m13.8689s               |
| 900        | 0.1793s                     | 00h02m41.3521s               |
| 1000       | 0.1788s                     | 00h02m58.8046s               |
| 5000       | 0.1500s                     | 00h12m29.8425s               |
| 10000      | 0.1196s                     | 00h19m56.0588s               |

we are able to classify all events from the previous day with no delay within the next 24 hours. In case of the volume of passengers gets closer to the original one before the outbreak (about 1 million), it is possible to configure extra servers to use parallel processing as detailed in Fig. 1.

![Fig. 5](image_url)  
**Fig. 5** The number of pictures taken per day inside the public transportation at Salvador. Each bin represents the total of passengers once four pictures are taken when it crosses the bus turnstile.
5 Results

This section discusses all results obtained by evaluating the pipeline steps detailed in Section 3: (i) filtering – Section 5.1; (ii) face detection – Section 5.2; and (iii) mask detection – Section 5.3.

5.1 Filtering

Aiming at assessing the methods considered in this step, we have selected 121,749 images taken on December 10th, 2019 between 5 am and 7 pm. Firstly, we run the CLAHE method to improve image quality. Next, we filter them to remove those whose illumination might affect the face detection, as discussed in Section 3.1. Figure 6 shows images to exemplify the three illumination conditions analyzed in this preprocessing step.

The left-most picture illustrates a situation with low luminosity. In this case, the detection of faces is complicated even when it is performed by specialists. Just below it, one may notice the histogram bins is concentrated between the interval [0, 50] as a unimodal distribution. The opposite situation is shown in the right-most picture, which affects not only face detection but also the mask classification. By looking at its histogram, we also observe a unimodal distribution shifted right. Both pictures can not be precisely analyzed and might add further misinterpretation on the adoption of masks. The best scenario is illustrated in the central picture, whose histogram resembles a multimodal distribution with bins distributed on the $i = [65, 210]$.

Table 2 shows the sum of every pixel inside the intervals presented in the histograms of Fig. 6. These values illustrate the steps considered in our empirical analysis to remove images with illumination problems. By using these interval thresholds, also validated by specialists, we removed 28.24% due to the low luminosity condition and 0.15% for presenting high luminosity, thus remaining a total of 87,182 (71.61%) images to be processed by our mask detection approach.

The evaluation of the two strategies designed to remove images with low quality (Section 3.2) was performed by creating a subset with 3,790 images, containing 1,105 with low illumination, 1,105 in good conditions, and 1,580 with high illumination. The experiment was organized as binary classification problem: good and bad condition. The
Table 2  Illumination distribution for every picture shown in Fig. 6 (in bold, the histogram intervals with the highest illumination values)

| Image type            | Intervals |
|-----------------------|-----------|
|                        | $i_1$     | $i_2$ | $i_3$ |
| Low luminosity        | **306565**| 628   | 7     |
| Normal luminosity     | 65723     | **177262** | 64215 |
| High luminosity       | 2779      | 39797 | **264624** |

comparison was performed by collecting the description time, the classification time, the total time, and final accuracy. For the ML models, the design of experiments was conducted considering a 10-fold cross validation and the parameters discussed in Section 3.2.

The results presented in Table 3 were organized in different set of experiments. In the first one, we have only used the histogram range, as discussed in the beginning of this section. The next ones show the results after combining different convolutional filters (LBP, Edge, and Emboss) along with the Random Forest (RF), Multilayer Perceptron (MLP), Support Vector Machine (SVM) models. According to this table, the best accuracy results were obtained by the combination between the Edge and Emboss filters and SVM. In our environment, we decided to use Emboss filter and SVM due to its lowest execution time.

5.2 Face detection

In this section, we summarize all results obtained with the four face detectors discussed in Section 3.3: ResNet, MTCNN, Faster R-CNN, and YOLO. Instead of retraining those models, we decided to use their pre-trained version as usually performed in the literature. Therefore, strategies such as retraining or zeroing weights were out of the scope of the public transportation company, which would like to use a classification model as sooner as possible. Firstly, we have randomly selected 300 images from the resultant dataset preprocessed by the previous layer. Next, specialists individually and manually have drawn bounding boxes, thus highlighting all faces to be later used as ground truth in our experiments.

Table 3  Comparison among different preprocessing strategies (the best model is highlighted in bold)

| Descriptor | Classifier | Feature extraction time (s) | Classification time (s) | Total time (s) | Accuracy |
|------------|------------|-----------------------------|-------------------------|----------------|----------|
| 1          | –          | –                           | –                       | 0.012          | 76%      |
| 2          | LBP        | RF                          | 0.05                    | 0.08           | 67%      |
|            | LBP        | MLP                          | 0.03                    | 0.03           | 63%      |
|            | LBP        | SVM                          | 0.04                    | 0.04           | 67%      |
| 3          | Edge filter| RF                          | 0.01                    | 0.03           | 73%      |
|            | Edge filter| MLP                          | 0.014                   | 0.019          | 81%      |
|            | Edge filter| SVM                          | 0.014                   | 0.016          | 86%      |
| 4          | Emboss filter| RF                          | 0.009                   | 0.034          | 82%      |
|            | Emboss filter| MLP                          | 0.008                   | 0.014          | 87%      |
|            | Emboss filter| SVM                          | 0.008                   | **0.011**      | **86%** |
Table 4 shows all classification results using the metric IOU defined in Section 3.5, which is widely adopted in several face detection studies. In the first line, we present the results from the application of every individual face detector. As one may notice, their performances vary from 45% up to 80%. We recall here all models were pre-trained and those results are only related to a test fold with 300 random images.

These IOU results point out the complexity of our scenario, i.e., even using pre-trained models with great results previously published in the literature, the best classification was achieved with MTCNN (80%). This drawback strongly affects our analysis, since an image with no face detected is automatically removed from our database. Therefore, any improvement in face recognition means that the more the detected faces, the fewer is the number of disregarded images.

Motivated by this situation, we have designed a new face-detector ensemble that aims at taking advantage of different methods. Lines 2–4, in Table 4, show the IOU results obtained by our ensembles. By combining all individual detectors (line 2), the classification result was 78%. Although it is lower than MTCNN, the IOU result was greater than the other ones. In line 3, we have selected the three best models, whose result was better than our first ensemble. Finally, we have combined the top 2 detectors, which provided the best results, thus overcoming the individual IOU value from MTCNN. Therefore, as expected, the best combination of our face-detector ensemble was obtained using the models that individually presented the best overall performance.

Aiming at better illustrating the execution of our ensemble, we have selected an image from our dataset as shown in Fig. 7. The left-most image shows the four bounding boxes estimated by the face detectors used in this work (line 1 of Table 4): ResNet in white, MTCNN in red, Faster R-CNN in yellow, and YOLO in green. Figure 7(B) shows the bounding box drawn by our ensemble after combining all detectors (line 2 of Table 4). The ensembles from Lines 3 and 4 of Table 4 yielded the bounding boxes illustrated in Fig. 7C and D, respectively.

In Fig. 8, we illustrate scenarios in which the users’ faces are not completely visible. By analyzing those faces detected by our ensemble, one may notice the bounding boxes were correctly drawn even in situation when the users were wearing glasses and cap. Another important result is presented in Fig. 8E which does not present a frontal photo of the user’s face. Finally, we see in Fig. 8B and C examples of multiple face detection and their classification as “wearing mask”. In Fig. 8B, our approach detected a second face which was cropped and partially covered by an arm.

| Models | IOU |
|--------|-----|
| 1      | ResNet 67% |
|        | MTCNN 80% |
|        | Faster R-CNN 45% |
|        | YOLO 74% |
| 2      | ResNet + MTCNN + Faster R-CNN + YOLO 78% |
| 3      | ResNet + MTCNN + YOLO 79% |
| 4      | MTCNN + YOLO 82% |
Fig. 7 (A) Bounding boxes defined by the four face detectors analyzed in our work: ResNet (white), MTCNN (red), Faster R-CNN (yellow), and YOLO (green). (B) Bounding box drawn by combining all detectors in our ensemble (line 2 of Table 4). The outputs produced by our ensembles presented in lines 3 and 4 of Table 4 are shown in (C) and (D).

Fig. 8 Images selected to illustrate the performance of our ensemble by detecting faces with occlusions. Images (A–D) and (F) show users wearing cap and glasses, whereas Image (E) illustrates a user who is not directly looking at the camera.
5.3 Mask detection

The last experiments were performed to assess our CNN architecture (Fig. 4 in Section 3.4) developed to predict, by using our ensemble, whether users were wearing masks. The design of experiments was performed considering the dataset presented in Section 3.4, which was split in training (90% – 9,816 images) and test (10% – 443 images) sets. The training step was executed using the 10-fold cross validation with a validation rate of 10%. The images selected to be part of every fold were randomly chosen in every 25 epoch, whose results are shown in Fig. 9.

In Fig. 9A, we notice an agreement between the accuracies (see (3)) from the training (red line) and validation (blue line) sets. Such a behavior along with the increasing accuracies emphasizes our architecture is, indeed, generalizing, i.e., the model has learned from the training set. Figure 9B ratifies this observation once the loss reduces, in both training (red line) and validation (blue line) sets, as the accuracy increases in consecutive epochs.

Table 5 shows the accuracy obtained by validating of every training fold. Although the best model was estimated from Fold 3, a stability is noticed in all folds with mean and standard deviation equal to 91.91% (0.84%).

After obtaining the best model, we have finally assessed our CNN architecture with 443 images, in which specialists found faces and classified them as: 198 wearing masks and 245 with no mask at all. In this experiment, the full execution of our approach (see Fig. 1) has provided outstanding results. According to the ROC curve presented in Fig. 10A, the true positive rate is maximized (4), whereas the false positive rate is minimized (5), thus leading an Area Under the ROC Curve (AUC) equals to 0.98. The overall error is better seen in the confusion matrix presented in Fig. 10B, which shows all faces were correctly classified, but 10 images misclassified as False Negative and 20 images as False Positive.

5.4 Visual interpretation

In this section, we present a set of visual metaphors created to provide a user-friendly tool to support the definition of more effective policies to deal with the COVID-19 pandemic. The first visualization, shown in Fig. 11, summarizes the rate of users wearing masks inside all buses per day. The left-most plot in this figure shows the rate on April 13th, 2020, which was the beginning of the pandemic period in Brazil, just before the local decree imposing the usage of masks. In the right-most one, we notice how the mask usage has

![Fig. 9](A) Accuracy and (B) loss obtained during each training epoch
Table 5. Accuracy values obtained in every training/validation fold (the highest accuracy is shown in bold).

| Folds | Accuracy  |
|-------|-----------|
| 1     | 92.77%    |
| 2     | 91.64%    |
| 3     | 93.00%    |
| 4     | 91.19%    |
| 5     | 92.55%    |
| 6     | 92.09%    |
| 7     | 91.42%    |
| 8     | 90.51%    |
| 9     | 91.42%    |
| 10    | 90.51%    |
significantly increased during the worst pandemic period in Brazil, on July 13th, 2020, when the transmission rate was $R_t > 1$ according to the Imperial College London\textsuperscript{5}.

Figure 12 expands the previous information over time. By looking at this plot, we can notice how the absolute number of people wearing masks has been varying. We call attention to a small number of users that, even knowing the risks of the virus and all alerts from the WHO, ignores the recommendation of wearing masks.

The two previous figures are useful to provide an overview of Salvador. Aiming at better understand, in detail, the users’ behavior in every line, we created individual plots as shown in Fig. 13. We can notice the same previous behavior in such plots, however, one can monitor the lines responsible for any general change.

Finally, in Fig. 14, we have selected a few lines to show the mask usage in different regions in Salvador. This visualization is strongly important to identify specific regions, by using a heatmap, in which policies need some reinforcement. Figure 14A shows the adoption of masks on April 13th, 2020, in the beginning of the COVID-19 pandemic, and on July 13th, 2020, after the local decree.

Besides these analyses presented in this manuscript, our results also allow us to proceed with such analyses at different hours and regions. Although our study was conducted on private datasets and systems designed to support local governments and public transportation companies, we have made available a short version of our system to illustrate our system as the reader can access at https://covid.neodados.com/.

6 Final remarks

This work presents an approach devoted to solving a practical issue caused by the COVID-19 outbreak. While no drug is available to cure or prevent SARS-CoV-2 infections, non-pharmacological strategies are our main mechanism to deal with the virus. In this sense, the World Health Organization (WHO) and scientists have agreed that, by wearing face masks, one avoids direct contact with the virus and, consequently, being a transmission vector in his/her community.

\textsuperscript{5}The transmission rate equals to one means every infected patient is contaminating another person. In Brazil, the worst rate was 2.81, i.e., every 100 infected was transmitting to other 281 persons.
Fig. 11  Evolution of the mask usage percentage in public transportation at different epochs

Fig. 12  The number per day of passengers that used masks or not

Fig. 13  Total usage of masks by line during specific days
In our scenario, in the city of Salvador in Brazil, public transportation plays an important role in the population’s daily routine. Millions of passengers take buses by making almost impossible to assure, in practice, other non-pharmacological strategies as, for example, a limited number of passengers and their minimum distance. This situation is especially complicated during the rush hour, once the local trade and economy started opening after about 8 months of quarantine and social isolation.

Our solution was designed by using Artificial Intelligence to analyze images taken from all passengers inside buses and predict whether or not they are wearing masks. To reach our goals, we have adopted several Deep Neural Network (DNN) architectures, mostly based on Convolutional Neural Network (CNN), previously published in the literature to solve similar issues. However, due to the complex environment (e.g. bus in movement, different illumination conditions, photo devices with low quality, and picture automatically taken without requiring a specific position), we were impelled to create a new ensemble of face detectors, thus providing a contribution to the state of the art in addition to practical usage of DNNs. Therefore, our contribution increases the number of faces detected and makes the prediction of mask usage more precise.

As a consequence, policymakers take advantage of our results to understand and visualize important information related to the rate of passengers wearing masks as, for instance, the time, days, and lines in which this rate reduces. Thus, advertisements and awareness work are performed with more effective results. Finally, it is important to highlight that our full approach is now a permanent solution that helps our population to struggle with not only COVID-19 and further pandemics but also epidemic and endemic diseases.

A possible limitation of our approach is the presence of users wearing transparent masks, which might increase the number of False Negative classifications. Although such images were not noticed in our experiments, we believe this is an important future work, which may require the design of a new ANN architecture and specific experimental setup. Another import future work is the detection of users who are not correctly wearing masks by placing them under mouth and nose as illustrated in Fig. 15. A possible solution for this problem might be obtained by, first, detecting the presence of masks in the image, and, then, the identification of nose and/or mouth inside the ROI.

As a final discussion, we understand that the adoption of pre-trained models has some limitations, such as the need for a greater amount of computation resources devoted to performing a feature extraction step on large datasets. Aiming at improving the approach...
proposed in this work, we plan to use model acceleration methods [47] to compress these pre-trained models without affecting the accuracy reduction.

Acknowledgments This work was partially supported by CAPES (Coordination for the Improvement of Higher Education Personnel – Brazilian Federal Government Agency). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of CAPES, and NVIDIA.

Declarations

Conflict of Interests The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest, or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: Large-scale machine learning on heterogeneous systems, https://www.tensorflow.org/. Software available from tensorflow.org
2. Bradski G (2000) The OpenCV Library. Dr. Dobb’s Journal of Software Tools
3. Bradski G (2000) The OpenCV Library. Dr. Dobb’s Journal of Software Tools
4. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
5. Canário JP, Mello R, Curilem M, Huenupan F, Rios R (2020) In-depth comparison of deep artificial neural network architectures on seismic events classification. J Volcanol Geotherm Res:106881
6. Celery: Distributed task queue, https://docs.celeryproject.org/. Accessed: 2021-03-26
7. Centers for Disease Control and Prevention (CDC) (2020). Symptoms of coronavirus disease 2019 (covid-19). https://www.cdc.gov. Online; Accessed November 03rd, 2020
8. Chu DK, Akl EA, Duda S, Solo K, Yaacoub S, Schünemann HJ, El-harakeh A, Bognanni A, Lotfi T, Loeb M et al (2020) Physical distancing, face masks, and eye protection to prevent person-to-person transmission of sars-cov-2 and covid-19: a systematic review and meta-analysis. The Lancet
9. Davies NG, Kucharski AJ, Eggo RM, Gimma A, Edmunds WJ, Jombart T, O’Reilly K, Endo A, Hellewell J, Nightingale ES et al (2020) Effects of non-pharmaceutical interventions on covid-19 cases, deaths, and demand for hospital services in the uk: a modelling study. The Lancet Public Health

Fig. 15 Examples of users wrongly wearing masks
10. Dong X, Yu Z, Cao W, Shi Y, Ma Q (2020) A survey on ensemble learning. Front Comput Sci 14(2):241–258
11. Fastapi framework, https://fastapi.tiangolo.com/. Accessed: 2021-03-26
12. Hanley JA, McNeil BJ (1982) The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology 143(1):29–36. PMID: 7063747
13. Harsoyo A, Rezi MC, Rusmin PH (2013) Design of face recognition system using local binary pattern and clahe on smart meeting room system. In: 2013 IEEE 3rd International Conference on System Engineering and Technology, pp 341–345
14. Haykin S (1994) Neural networks: A comprehensive foundation, 1st edn. Prentice Hall PTR, Upper Saddle River
15. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
16. Heikkilä M, Pietikäinen M, Schmid C (2009) Description of interest regions with local binary patterns. Pattern Recogn 42(3):425–436
17. Hendrawan A, Asmiatun S (2018) Identification of picnosis cells using contrast-limited adaptive histogram equalization (clahe) and k-means algorithm. In: 2018 1st International Conference on Computer Applications Information Security (ICCAIS), pp 1–3
18. Hitam MS, Awalludin EA, Yussof WNJHW, Bachok Z (2013) Mixture contrast limited adaptive histogram equalization for underwater image enhancement. In: 2013 International conference on applications technology (ICCAT). IEEE, pp 1–5
19. Jain V, Learned-Miller E (2010) Fddb: A benchmark for face detection in unconstrained settings, Technical report. UMass Amherst technical report
20. Kharel N, Alsadoon A, Prasad PWC, Elchouemi A (2017) Early diagnosis of breast cancer using contrast limited adaptive histogram equalization (clahe) and morphology methods. In: 2017 8th International Conference on Information and Communication Systems (ICICS), pp 120–124
21. King DE (2009) Dlib-ml: A machine learning toolkit. J Mach Learn Res 10(60):1755–1758
22. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
23. Kolhar M, Al-Turjman F, Alameen A, Abualhaj MM (2020) A three layered decentralized iot biometric architecture for city lockdown during covid-19 outbreak. IEEE Access 8:163608–163617
24. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
25. Kumar A, Kaur A, Kumar M (2019) Face detection techniques: a review. Artif Intell Rev 52(2):927–948
26. Lateef F, Ruiychek Y (2019) Survey on semantic segmentation using deep learning techniques. Neuro-computing 338:321–348
27. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2323
28. Loey M, Manogaran G, Taha MHN, Khalifa NEM (2021) A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic. Measurement 167:108288
29. Luo D, Wen G, Li D, Hu Y, Huan E (2018) Deep-learning-based face detection using iterative bounding-box regression. Multimed Tools Appl 77(19):24663–24680
30. Markel H, Lipman HB, Navarro JA, Sloan A, Michalsen JR, Stern AM, Catron MS (2007) Nonpharmaceutical interventions implemented by us cities during the 1918-1919 influenza pandemic. JAMA 298(6):644–654
31. Matthew D, Fergus R (2014) Visualizing and understanding convolutional neural networks. In: Proceedings of the 13th European Conference Computer Vision and Pattern Recognition, Zurich, Switzerland, pp 6–12
32. Merkel D (2014) Docker: lightweight linux containers for consistent development and deployment. Linux J 2014(239):2
33. Ng H-W, Winkler S (2014) A data-driven approach to cleaning large face datasets. In: 2014 IEEE international conference on image processing (ICIP). IEEE, pp 343–347
34. Nieto-Rodríguez A, Mucientes M, Brea VM (2015) System for medical mask detection in the operating room through facial attributes. In: Paredes R, Cardoso JS, Pardo XM (eds) Pattern Recognition and Image Analysis. Springer International Publishing, Cham, pp 138–145
35. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
36. Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. British Machine Vision Association
37. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
38. Qin B, Li D (2020) Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19. Sensors 20(18)
39. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
40. Ren S, He K, Girshick R, Sun J (2017) Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
41. Schünemann HJ, Akl EA, Chou R, Chu DK, Loeb M, Lotfi T, Mustafa RA, Neumann I, Saxinger L, Sultan S et al (2020) Use of facemasks during the covid-19 pandemic. Lancet Respiratory Med 8(10):954–955
42. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
43. Singh R, Om H (2017) Newborn face recognition using deep convolutional neural network. Multimed Tools Appl 76(18):19005–19015
44. Ud Din N, Javed K, Bae S, Yi J (2020) A novel gan-based network for unmasking of masked face. IEEE Access 8:44276–44287
45. Vapnik V (2013) The nature of statistical learning theory. Springer science & business media
46. Viola P, Jones M (2001) Robust real-time face detection. In: null. IEEE, p 747
47. Wang X, Zheng Z, He Y, Yan F, Zeng Z, Yang Y (2021) Soft person reidentification network pruning via blockwise adjacent filter decaying. IEEE Trans Cybern:1–15
48. Wu X, Sahoo D, Hoi SCH (2020) Recent advances in deep learning for object detection. Neurocomputing 396:39–64
49. Xiang J, Zhu G (2017) Joint face detection and facial expression recognition with mtcnn. In: 2017 4th International Conference on Information Science and Control Engineering (ICISCE). IEEE, pp 424–427
50. Yang S, Luo P, Loy CC, Tang X (2016) Wider face: A face detection benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
51. Yang W, Zhou L, Li T, Wang H (2019) A face detection method based on cascade convolutional neural network. Multimed Tools Appl 78(17):24373–24390
52. Zafeiriou S, Zhang C, Zhang Z (2015) A survey on face detection in the wild: Past, present and future. Comput Vis Image Underst 138:1–24
53. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503
54. Zuiderveld K (1994) Contrast limited adaptive histogram equalization. In: Graphics Gems IV. Academic Press Professional, Inc., USA, pp 474–485

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
Affiliations

João Paulo Canário\textsuperscript{1} · Marcos Vinícius Ferreira\textsuperscript{2} · Junot Freire\textsuperscript{2} · Matheus Carvalho\textsuperscript{3} · Ricardo Rios\textsuperscript{1}

João Paulo Canário
joao.canario@ufba.br

Marcos Vinícius Ferreira
marcos.ferreira@neodados.com

Junot Freire
junot@neodados.com

Matheus Carvalho
matheus.souza@gevan.com.br

\textsuperscript{1} Departament of Computer Science, Federal University of Bahia, Salvador, Brazil
\textsuperscript{2} Neodados, Smart Solutions, Salvador, Brazil
\textsuperscript{3} Integra, Association of Public Transportation Companies, Salvador, Brazil