applied sciences

MDPI

Article

A Computer Vision Model to Identify the Incorrect Use of Face Masks for COVID-19 Awareness

Fabricio Crespo [1,2], Anthony Crespo [1], Luz Marina Sierra-Martínez [3,4], Diego Hernán Peluffo-Ordóñez [4,5,6] and Manuel Eugenio Morocho-Cayamcela [1,2,4,*]

1 School of Mathematical and Computational Sciences, Yachay Tech University, Urcuquí 100119, Ecuador; jonnathan.crespo@yachaytech.edu.ec (F.C.); brian.crespo@yachaytech.edu.ec (A.C.)
2 Deep Learning for Autonomous Vehicles, Robotics, and Computer Vision (DeepARC Research), Urcuquí 100115, Ecuador
3 Information Technology R&D Research Group—GTI, University of Cauca, Popayán 190003, Colombia; lsierra@unicauca.edu.ec
4 SDAS Researh Group, Ben Guerir 43150, Morocco; diego.peluffo@sdas-group.com
5 Modeling, Simulation and Data Analysis (MSDA) Research Program, Mohammed VI Polytechnic University, Benguerir 43150, Morocco; peluffo.diego@um6p.ma
6 Faculty of Engineering, Corporación Universitaria Autónoma de Nariño, Pasto 520001, Colombia; diego.peluffo@aunar.edu.co
* Correspondence: mmorocho@yachaytech.edu.ec or manuel.morocho@sdas-group.com

Abstract: Face mask detection has become a great challenge in computer vision, demanding the coalition of technology with COVID-19 awareness. Researchers have proposed deep learning models to detect the use of face masks. However, the incorrect use of a face mask can be as harmful as not wearing any protection at all. In this paper, we propose a compound convolutional neural network (CNN) architecture based on two computer vision tasks: object localization to discover faces in images/videos, followed by an image classification CNN to categorize the faces and show if someone is using a face mask correctly, incorrectly, or not at all. The first CNN is built upon RetinaFace, a model to detect faces in images, whereas the second CNN uses a ResNet-18 architecture as a classification backbone. Our model enables an accurate identification of people who are not correctly following the COVID-19 healthcare recommendations on face mask use. To enable further global use of our technology, we have released both the dataset used to train the classification model and our proposed computer vision pipeline to the public, and optimized it for embedded systems deployment.

Keywords: artificial intelligence; deep learning; computer vision; face mask recognition; object detection; image classification; COVID-19

Citation: Crespo, F.; Crespo, A.; Sierra-Martínez, L.M.; Peluffo-Ordóñez, D.H.; Morocho-Cayamcela, M.E. A Computer Vision Model to Identify the Incorrect Use of Face Masks for COVID-19 Awareness. *Appl. Sci.* **2022**, *12*, 6924. https://doi.org/10.3390/app12146924

Academic Editor: Hui Yuan

Received: 10 May 2022
Accepted: 28 June 2022
Published: 8 July 2022

1. Introduction

Human-to-human transmission plays an essential role in the spread of severe acute respiratory syndrome coronavirus-2 (SARS-CoV-2, 2019-nCoV), causing the disease named COVID-19 [1,2]. It is known that one of the principal ways of SARS-CoV-2 transmission is through tiny droplets ejected from a pre-symptomatic patient while sneezing, coughing, or just speaking. Therefore, the use of face masks was recommended by the World Health Organization (WHO), especially in public areas, to reduce the rates of virus spreading [3]. The mandatory use of face masks varies over countries; for instance, due to the arrival of COVID-19 vaccines and the low contagion rate, some countries such as Israel have waived obligatory face mask use [4]. However, some countries still have high rates of COVID-19 cases, and a significant part of the population is not vaccinated yet, making the use of face masks a mandatory rule [5–8]. These countries are facing the challenge of creating a preventive system to detect the incorrect use (or the absence) of face masks.

In general, monitoring people who do not respect the use of face masks is a challenging task, especially in crowded public areas where it becomes impossible to track whether every-

one is using face masks according to the WHO recommendations. For this reason, there is great need for a computer system that enables the automatic monitoring of correct/incorrect face mask use. Currently, computer vision (CV) tasks are useful for solving problems related to object detection, classification, object counting, visual surveillance, etc., taking advantage of video resources from public surveillance cameras located in many public areas (i.e., shopping malls, supermarkets, airports, train stations, stadiums, etc.) [9–12]. The problem of the correct/incorrect wearing of face masks implies two CV tasks: (1) object detection and (2) object classification. The object detection task is helpful for finding the faces of people in images or videos, and the object classification task divides the faces detected by the object locator into different classes (e.g., correct or incorrect use of face masks).

CV is an exciting approach that provides the necessary tools to build systems capable of detecting and classifying the correct/incorrect use of face masks. However, to increase the accuracy of the detection and classification models, CV researchers are needed to help in the development of models that enhance the existent systems. In the last year, researchers in the CV community have worked hard to propose deep learning (DL) models to tackle this vital area for the benefit of society's healthcare [13–16]. However, a significant part of the research is focused on generating suitable datasets to train the models, especially for the detection and classification of occluded faces. Therefore, it is important to address various approaches to the classification for COVID-19 face mask wearing in order to produce an accurate DL model.

In this work, we propose a compound convolutional neural network (CNN) pipeline for COVID-19 face mask detection and classification. In addition, by conducting a grid search among the optimizer and cost function hyperparameters, we present the best configuration that optimizes the model in terms of classification accuracy. Moreover, a new cleaned dataset is presented to train the object classification model. To enable reproducible research, we have used the public and open Face Mask Label Dataset (FMLD) [17] and Medical Mask Dataset (MMD) [18] to train the classifier with distinct DL models.

2. Related Work

As soon as the pandemic hit the globe, research studies related to the generation of datasets containing images of face masks were released to the CV community. However, even before the pandemic, there was an active research field studying the detection of face masks and related dataset formation. In addition, some papers gave special attention to occluded faces (i.e., detecting any face with accessories such as sunglasses and face masks, and partially visible faces). In the following subsections, related studies on the topic are outlined.

2.1. Face and Face Mask Datasets

2.1.1. Face Datasets

These are specialized datasets used to train models that detect faces in images or frames of a video. *WiderFace* is one of the most extensive datasets oriented to CV tasks on human faces [19]. The dataset was presented in 2016, and it contains 32,203 images with 393,703 face labels. It inherits images from the extensive *Wider* dataset. Another important dataset is *Face Detection Data Set and Benchmark* (FDDB) [20]. FDDB was published in 2010, and it contains annotations from 5171 faces in 2845 images. The dataset was generated using real-life scenarios such as occluded faces. FDDB has been one of the pioneer datasets in the topic. *Annotated Faces in the Wild* (AFW) is composed of 205 images with 468 faces. It was released in 2012 as a part of the work in face detection and pose estimation [21].

2.1.2. Face Mask Datasets

Researchers have already addressed an important challenge in face detection for occluded faces and face forensics [22–24]. Years before the pandemic, the CV community had been trying to generate a standard dataset oriented to accomplishing this task, without success. Notwithstanding, one of the most relevant datasets, called *MAFA*,

was published in 2017, with 30,811 images downloaded from the Internet, containing 35,806 masked face annotations [25]. On each image, at least one face is occluded with a mask. The COVID-19 pandemic demands that special attention be paid to creating new datasets representing modern COVID-19 face mask scenarios, either real or virtual. In this connection, *MaskedFace-Net* was released in 2021 to solve the problem of the lack of large mask datasets [26]. It comprises 137,016 editable images (i.e., masks added to faces through photo editing programs), with two classes: "correctly masked dataset" and "incorrectly masked face dataset". Additionally, in [27], three datasets were introduced: the *Masked-Face Detection Dataset* (MFDD), the *Real-World Masked Face Dataset*, and the *Simulated Masked Face Recognition Dataset* (SMFRD), with 24,771, 95,000, and 500,000 images, respectively. MFDD contains images crawled from the internet, with labels only if the subjects are wearing masks. RMFRD is the world's largest dataset of real-world masked faces, according to the authors. The dataset contains 5000 images, with 525 images of people using masks and 90,000 images of real people without masks. The dataset is free and available in GitHub. However, the dataset also includes some faces wearing masks inappropriately. Hence, a future classification to train the model in order to recognize this new class might be developed. Furthermore, SMFRD contains real face images with facial masks added artificially to simulate faces wearing masks. Only one part of this dataset is publicly available for download. Finally, the *Moxa3K* dataset [28] contains images captured from Russia, Italy, China, and India during the pandemic. The dataset has 3000 images in total. The numbers of faces that can be extracted from the dataset are 9161 without masks and 3015 masked.

2.2. Relevant Face Detection Models

Through the years, multiple research studies have been presented on face detection. One of the most influential studies is the paper presented by Paul Viola and Michael Jones [29]. Their model was based on rapid object detection using a boosted cascade of simple features. The Viola and Jones proposal is known worldwide for being the first CV model that employed a rudimentary machine learning (ML) technique known as boosted decision trees. The central idea was to create an algorithm capable of recognizing faces and drawing a box around them. With the advent of CNNs, the face detection error decreased significantly [30,31]. Subsequently, DL research accelerated its pace, and novel deep neural network architectures were proposed to tackle the face detection problem. The main advantage of using DL architectures is the increased accuracy of the models. In [32], the authors used a faster region-based CNN framework to perform face detection. This study obtained a state-of-the-art face detection performance on the FDDB benchmark. Another important model is RefineFace, which is an improved face detection model based on ResNet, with a 6-level feature pyramidal structure used as the base of the network [33]. RefineFace is also based on the face detector RetinaNet, with five new modules: STR, STC, SML, FSM, and RFE. The results obtained by varying the modules are presented in the paper, with accuracies near and above 90%. The necessity for a model to detect faces in a complex background and the large amount of time consumed are reasons why the authors in [34] presented face detection using improved TinyYOLOv3 and an attention mechanism. This is composed of an improved TinyYOLOv3 capable of extracting significant semantic information by changing the traditional convolution to deep separable convolution, i.e., dividing a single convolution into two or more parts to obtain a model of smaller size with a high detection speed. In addition, the attention mechanism was added to the feature extractor layers to improve the detection position. Lastly, a face detection algorithm based on a double-channel CNN with occlusion perceptron is a method that uses a VGG16 as a backbone, with the addition of a specialized unit to judge occluded areas, making it an occlusion perceptron neural network. The double channel refers to the capacity of the residual network to extract features of the whole face, while the perception neural network extracts the features of the occluded parts. Both results are combined to produce the final prediction. This method improves detection speed and accuracy [35].

2.3. Face Mask Detection and Classification for COVID-19

Multiple DL models were released from the beginning of the COVID-19 pandemic with different approaches to solving the face mask detection and classification challenges. For example, in [14], the authors propose a hybrid model using DL and classical ML. The first part uses a ResNet-50 architecture as a feature extractor, while decision trees and support vector machines are used to classify the positions of the masks. Similarly, in [15], the authors present a model capable of identifying face-mask-wearing conditions by combining super-resolution images and classification networks such as SRCNet. In addition, the use of ResNet-50 as a feature extractor and YOLOv2 as a detector has been recently proposed [8]. In [36], the authors proposed the so-called FMD-Yolo framework to detect whether people wear masks in public spaces correctly. This framework employs Im-Res2Net-101 and the path aggregation network En-PAN for the steps of feature extractor and feature fusion, respectively. The authors test their approach against several state-of-the-art detection algorithms using two publicly available datasets. Their results on both datasets outperformed the approaches to which they were compared. Context-Attention R-CNN is another detection method using a new framework for recognition of masked faces [37]. The method is based on multiple context feature extractor components, decoupling branch components, and attention components. In addition, the researchers created a dataset of 8635 faces under different experimental conditions. The *mean average precision* (mAP) for the model was 84.1% over the dataset previously mentioned, which was 6.8% mAP higher than Faster R-CNN. Finally, Face Mask Recognition Network (FMRN) is a detection model that uses an algorithm to classify images via posture recognition. Then, the network processes the images according to the classes [38].

3. Methodology

3.1. The Datasets Used to Train the Proposed Model

3.1.1. The WIDER FACE Dataset

This dataset was presented in the work "WIDER FACE: A Face Detection Benchmark" [19], and it was used by the authors of RetinaFace to produce a face detection model. WIDER FACE is a public dataset that has been previously used to train alternative face recognition models. An important feature of this dataset is the inclusion of challenging faces with different dimensions, poses, expressions, illumination, make-up, resolutions, scales, and occlusions. This fact closes the gap between real-world requirements and the available face datasets. In addition, WIDER FACE is one of the most extensive publicly available datasets and has become a benchmark for the face detection modeling task. Moreover, the dataset offers additional information such as bounding boxes coordinates, categories, poses, and occlusions. This dataset is based on the WIDER dataset, and it has 32,203 images with 393,703 labeled faces with different poses, occlusions, and sizes [39].

3.1.2. Face Mask Label Dataset (FMLD)

This dataset is used in our model to detect the correct or incorrect use of face masks from the images obtained from the face recognition task using WIDER FACE. FMLD was first presented in [17], where the authors released the dataset with bounding-box annotations around face masks. This dataset has labels for images with faces with (i) appropriate and (ii) inappropriate use of COVID-19 masks. It is worth noting that the masks used during the COVID-19 pandemic had different shapes, colors, models, and patterns, and it requires an updated dataset to cover all those instances. The FMLD dataset takes images from two donor datasets: MAFA [25] and WIDER FACE [19].

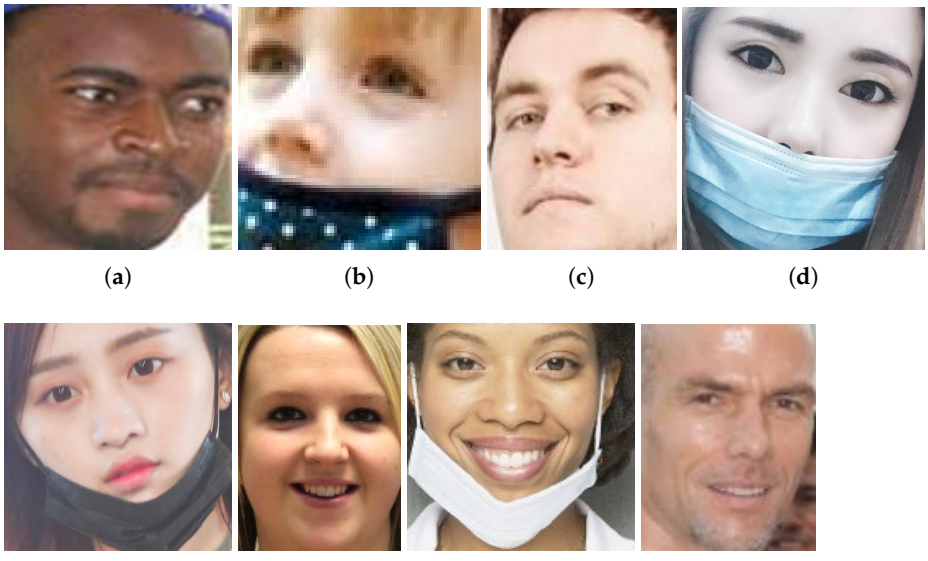
The MAFA dataset is the principal source of images of correctly and incorrectly worn face masks. In contrast, the WIDER FACE dataset is used as a source of different images of faces without masks. Therefore, the FMLD dataset contains a realistic set of images of faces with and without COVID-19 masks in different situations in real life. The FMLD dataset gives information such as the donor dataset, the folder and file name, the label (mask use or no mask use), and the coordinates of the pixels where the face is located.

Another important feature of the FLML dataset is the variety in terms of the gender of the subjects, the expressions of the faces, and the ethnicity of the subjects. This helps the DL models to learn a diversity of features present in real-world images. The FMLD dataset was used to train and test the different face mask detection DL models studied in this article. Figure 1 presents a subset of examples of the class "Compliant", where the correct use of a face mask is shown, whereas Figure 2 shows examples of the "Non-Compliant" class. Table 1 summarises the FMLD dataset construction.

(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 1. A subset of images taken from the "Compliant" class of the FMLD dataset. The examples (**a**–**h**) show faces with the correct use of masks according to the WHO advice.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 2. A subset of images taken from the "Non-Compliant" class of the FMLD dataset. The examples (**a**–**h**) show faces that are incorrectly using (or not using) masks.

Table 1. Summary of the construction of the FMLD dataset. FMLD uses the donor datasets MAFA and WIDER FACE to construct the "Compliant" and "Non-Compliant" classes.

Donor Dataset	Purpose	Images	Faces	Labels		
				With Mask		Without Mask
				Correct	Incorrect	
MAFA	Training	25,876	29,452	24,603	1204	3645
	Testing	4935	7342	4929	324	2089
WIDER FACE	Training	8906	20,932	0	0	20,932
	Testing	2217	5346	0	0	5346
FMLD	Training	34,782	50,384	24,603	1204	24,577
	Testing	7152	12,688	4929	324	7435
	Totals	41,934	63,072	29,532	1528	32,012

3.2. Detection and Classification Models

3.2.1. Face Detection: RetinaFace

This is also known as single-shot multi-level face localization in the wild. This DL model is a state-of-the-art technique for face detection that uses an end-to-end detector, and it performs three different face localization tasks together: (i) face detection, (ii) 2D face alignment, and (iii) 3D face reconstruction. RetinaFace uses the ResNet architecture plus fully pyramidal networks (FPN) to obtain a robust feature representation of the image. It outputs the bounding box for the face, five facial landmarks (denoting eyes, nose, and mouth), and a dense 3D mapping of points to represent the face [40].

3.2.2. Face Classification: ResNet

Since the revolution of CNNs with the successful introduction of AlexNet [41], researchers have tried to obtain more accurate models to work with CV image tasks. Nevertheless, making deeper CNNs does not only involve adding more layers, since problems such as the vanishing gradient might appear (when the gradient is backpropagated to previous layers, it can converge to an infinitely small number, causing low performance in the CNN). However, vanishing problems were solved in some studies using normalized initialization and intermediate normalization layers, allowing the models to converge for stochastic gradient descent (SGD) and backpropagation. Although they resolved the vanishing problem, He et al. [42] observed that the accuracy in the training dataset dropped in models with more layers. Therefore, a degradation problem had appeared. This problem leads to the saturation of the accuracy in the training dataset, but it is not caused by overfitting. He et al. presented ResNets as a solution to the degradation problem. They realized that the deeper layers were not identity mapping. This "identity mapping" was regarded as the ability to maintain at least the same error across the deeper layers, like an identity function avoiding degraded performance. Therefore, this is the desired underlying mapping. Instead of waiting for the layers to fit the desired underlying mapping, the authors explicitly allowed the layers to fit a residual mapping. These residual mappings are also known as residual blocks (identity blocks). Residual blocks skip one or more connection layers to enable a fine-grained level of detail at the inference. ResNet-18 is a variant of ResNet, and its architecture is presented in Figure 3. Here, we can see the shortcuts that contribute to identity mapping. Depending on whether or not there is a necessity to match dimensions through each residual block, two types of identities can be found: the identity shortcuts (solid lines) and the projection shortcuts to create a match between the input and output dimensions (dotted lines). The matching of dimensions can be performed using linear projections.

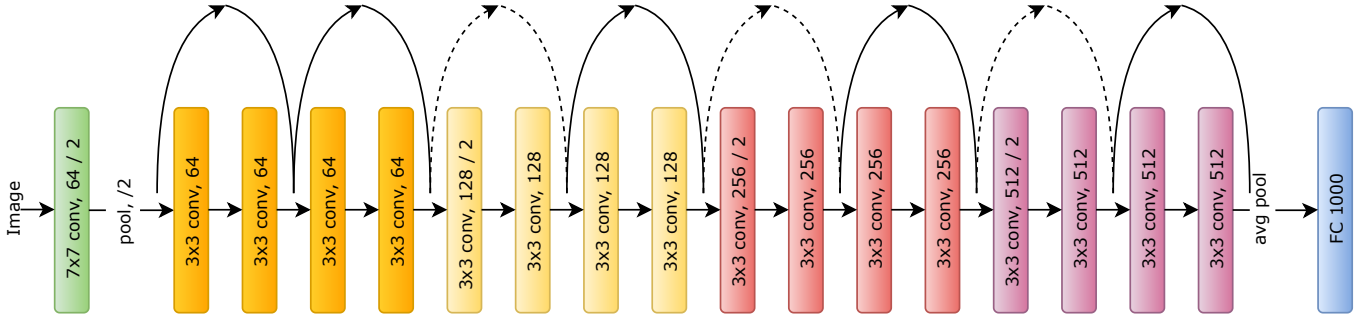


Figure 3. ResNet-18 architecture composed of residual blocks with two types of shortcuts: identity shortcuts and projection shortcuts.

3.2.3. Face Classification: ResNeSt

This variant of the original ResNet takes advantage of the introduction of feature map attention and multi-path representation, which are both essential techniques for visual recognition. In addition, ResNeSt adds split-attention blocks (i.e., computational units composed of feature groups and split-attention operations). As a result, ResNeSt presented better transfer learning results when used as a backbone on many public benchmarks [43].

3.3. The Two-Stage Pipeline CNN

The proposed pipeline is composed of (1) a face detection stage, whose output is the input for the (2) face classification stage, to detect the use of COVID-19 face masks. According to the mask placement, the pipeline will take images or frames and classify them into "Compliant" and "Non-Compliant". The pipeline deployment was tested, keeping RetinaFace as the predefined detection model and changing the classification models, specifically the ResNet and ResNeSt variants. In addition, the inference behavior of all the models was studied under different optimization algorithms.

3.4. Loss Function and Optimizers

Let $f(\cdot)$ be the proposed classification function, which takes the matrix of n input images $\boldsymbol{X} \in \{X^{(i)} \dots X^{(n)}\}$ and the weight matrix $\boldsymbol{W}$ as inputs and returns the predicted label represented as $\hat{Y} \in \{\texttt{compliant}, \texttt{non-compliant}, \texttt{incorrect}\}$. Therefore, we can denote the estimated label as follows:

$$\hat{\boldsymbol{Y}} = f(\boldsymbol{X}, \boldsymbol{W}). \tag{1}$$

To measure the performance of our classification model, the total loss $\mathcal{L}_T$ between the ground truth and the distribution $\hat{I}$ can be computed with the *cross-entropy* loss. The classification network iteratively updates the values of $\boldsymbol{W}$ by backpropagating the error through the neural network and converging to the local minimum value of the total loss $\mathcal{L}_T$. The labeled images from the FMLD dataset can be formalized as

$$(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}) \quad \text{for} \quad i \in \{1, \dots, n\}, \tag{2}$$

where n represents the total number of images in the dataset, and $\boldsymbol{X}^{(i)}$ and $\boldsymbol{Y}^{(i)}$ represent the ith image and its estimated label, respectively.

Our model computes the weights $\boldsymbol{W} = \{W^{(1)}, \dots, W^{(n)}\}$ that correspond to the minimum value of the total loss $\mathcal{L}_T$ as follows:

$$\mathcal{L}_T = -\frac{1}{n} \sum_{i=1}^{n} L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{W}), \tag{3}$$

where $L(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{W})$ represent the loss in terms of the dataset and the parameters/weights. Equation (3) can be written in the logarithmic likelihood form as follows:

$$\mathcal{L}_T = -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{Y}^{(i)} log\left(\hat{p}^{(i)}\right), \tag{4}$$

with $\hat{p}^{(i)}$ as the estimation probability that the ith image matches the intended l value [44]. Note that $\hat{p}^{(i)}$ depends on the values of $\boldsymbol{W}$.

For the CNN optimization, we let the cross entropy with respect to W (Equation (4)) be represented as $\delta = \nabla_W \mathcal{L}(\boldsymbol{W})$, where δ represents the gradient. Therefore, we can write

$$\delta = \frac{1}{n}\nabla_W \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{X}^{(i)}, \boldsymbol{Y}^{(i)}, \boldsymbol{W}), \tag{5}$$

where δ represents the derivative of $\mathcal{L}(\boldsymbol{W})$.

Adagrad [45], Adam [46], stochastic gradient descent (SGD) [47], SGD with momentum [48], and RMSprop [48] were used to estimate the optimal set of values in $\boldsymbol{W}$ that minimizes $\mathcal{L}(\boldsymbol{W})$.

3.5. Framework and Hardware Acceleration

3.5.1. ML Framework

The framework employed to design and implement the pipeline was PyTorch, since all the pre-trained models under study are implemented in the framework, and they could be easily imported through *torchvision* .

3.5.2. Hardware Acceleration

On the hardware acceleration side, we used Tesla V100 and Tesla P100 graphical processing units (GPUs) with 16GB from Google Colaboratory, with a 2.00 GHz Intel ® Xeon(R) CPU. The GPUs play an important role in training a DL model because, depending on their capacity, the training can take more or less time. It is essential to know that the Tesla V100 GPU has a performance of 14,029 gigaflops while the Tesla P100 has 10,609 gigaflops of performance. In addition, the Tesla V100 is superior regarding the number of cores; it has 5120 cores and the Tesla P100 has 3584 cores. Therefore, it is clear that the Tesla V100 is more powerful than the Tesla P100.

3.6. Performance Measures

To validate the performance of the classification stage, the mean average precision *(mAP)* was used as a metric to determine the most appropriate model. Additional metrics were also used:

1. **Accuracy**: Percentage of correct predictions.

$$accuracy = \frac{\text{all correct predictions}}{\text{all samples}} \tag{6}$$

$$= \frac{TP + TN}{TP + FN + FP + TN} \tag{7}$$

2. **Precision**: The number of correct predictions.

$$precision = \frac{TP}{TP + FP} \tag{8}$$

3. **Recall**: The proportion of the true positives that were correct.

$$recall = \frac{TP}{TP + FN} \tag{9}$$

4. **F1-score**: The F1-score is the harmonic mean between the precision and recall. It is frequently used when there is an imbalance in the dataset. One crucial fact is that this metric considers how many errors the model has per class and their influence on the final performance of the model, and not only the absolute number of right or wrong predictions.

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision \cdot Recall} \quad (10)$$

3.7. Preparation of the Dataset

Dataset Splitting

In order to conduct the training, it is important to split the dataset into training, validation, and test datasets. The proportion for each dataset was 80% for training, 10% for validation, and 10% for testing.

4. Results and Discussion

4.1. Phase I: Testing Deep Learning Classification Models with Different Optimizers

In this section, the results obtained in the two-stage pipeline are presented. Special attention is given to the classification stage, since the pre-trained RetinaFace model performed the face detection part, requiring no fine-tuning. Therefore, the image classification task concerning COVID-19 masks was the pivotal part of this study. The principal objective was to gain an insight into the performance of the DL models while solving an urgent real-life problem. We implemented all the variants of ResNet and one variant of ResNeSt using transfer learning. The initial weights were borrowed from the pre-trained model with the ImageNet dataset. Then, all the models were trained using the FMLD dataset, with RBG input images of size $W \times H$ pixels as inputs, where W and H were ≥ 10 to avoid losing the activation maps during the pooling operations. The dataset was randomly divided into training, validation, and test sets. An overview of the resulting datasets can be found in Table 2.

Table 2. An overview of the dataset split used to train the different classification models.

Dataset	Classes	
	Compliant	***Non-Compliant***
Training	7028	17,525
Validation	1757	4382
Test	4795	137,50
Total	13,580	35,657

The cross-entropy was adopted as a loss function. At the same time, the following optimizers were considered: (1) SGD with a learning rate equal to 0.001 and a reduction factor of 0.1 after seven epochs, (2) SGD with a momentum value of 0.9, (3) Adam with a learning rate of 0.001, (4) RMSprop with a learning rate of 0.01, and (5) Adagrad with a learning rate of 0.01. The models were trained with 15 epochs, each with a batch size equal to 16. The behavior of the accuracy and loss during the training and validation stages of ResNet-34 with the Adagrad optimizer led to this being selected as the optimal configuration. The ResNet-34 with Adagrad results can be found in Figures 4–7. RMSprop, Adam, and SGD were less effective optimizers in either the training or validation datasets, since they resulted in the highest losses and lowest accuracy. On the other hand, SGD with momentum and Adagrad optimizers resulted in the highest accuracy and lowest loss, but their differences were minimal. The details are given as follows:

1. **ResNet-18 and ResNet-152:** The best performance was obtained with the Adagrad optimizer for both the training and validation datasets.
2. **ResNet-34, ResNet-50, ResNet-101, and ResNeSt-200:** In these models, the best result on the training dataset was obtained with the Adagrad optimizer, and for the validation dataset, SGD with momentum was more robust.

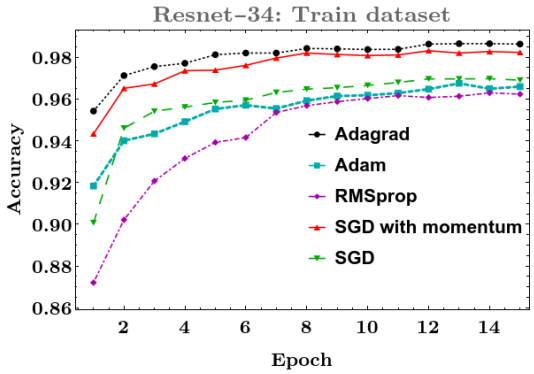


Figure 4. ResNet-34 accuracy on training dataset.

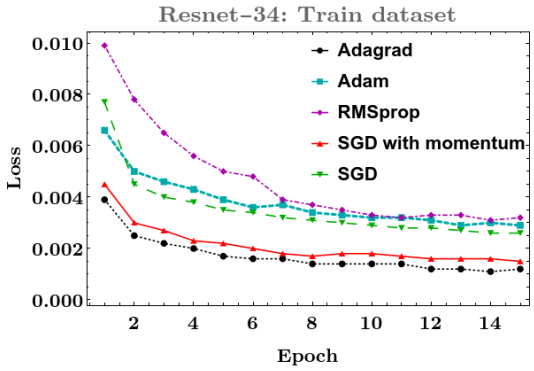


Figure 5. ResNet-34 loss on training dataset.

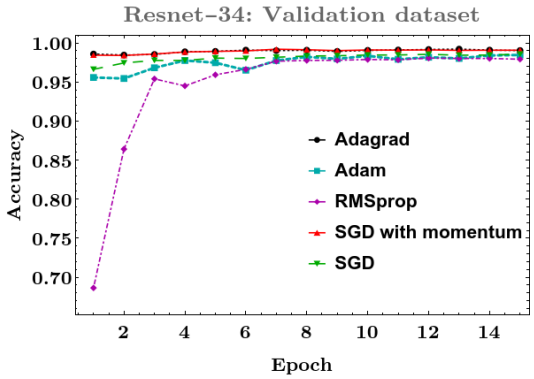


Figure 6. ResNet-34 accuracy on validation dataset.

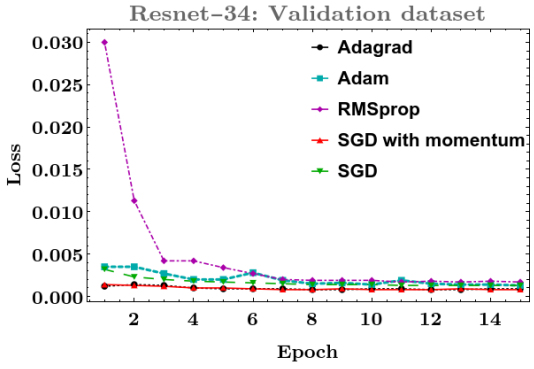


Figure 7. ResNet-34 loss on validation dataset.

However, it is necessary to analyze the results obtained on the test dataset to verify the performance of each model. Therefore, the experiments were performed using 18,545 images (4745 Compliant and 13,750 Non-Compliant), and the accuracy results are presented in Table 3.

Table 3. Models with different optimizers.

	Accuracy %				
Model	***SGD***	***RMSprop***	***Adam***	***SGD with momentum***	***Adagrad***
ResNeSt-200	96.44	92.41	95.79	**97.01**	96.78
ResNet-18	96.06	95.24	95.95	96.35	**96.64**
ResNet-34	96.06	95.28	95.86	96.4	**97.24**
ResNet-50	96.54	95.82	95.82	**96.59**	96.21
ResNet-101	95.64	94.95	95.29	**96.8**	96.63
ResNet-152	95.33	94.92	95.29	**97.21**	97.08

As can be seen, all the models achieved an accuracy greater than 94% except ResNeSt-200 with the RMSprop optimizer, which obtained 92.41%. The more accurate models are those trained with the SGD with momentum and Adagrad optimizers. ResNet-34 with Adagrad is the model configuration with the highest accuracy, followed by ResNet-152 with SGD with momentum and ResNet-152 with Adagrad. All the models, ordered by their accuracy percentages, can be seen in Figure 8. The lowest accuracy models are those trained with the RMSprop optimizer.

To analyze the results obtained in the test dataset, we chose ResNet-34 with the Adagrad optimizer as the model for performing the predictions. A confusion matrix is presented in Figure 9 to aid in understanding the results. A value of 0 represents the label "Compliant", and a value of 1 represents the class "Non-Compliant". According to Table 2, there were 4795 "Compliant" images, and the model found 4673 of them (with 122 wrong predictions). For the "Non-Compliant" class, the model found 13,360 from a total of 13,750 (with 390 wrong predictions). Therefore, the model obtained an accuracy performance of around 97%.

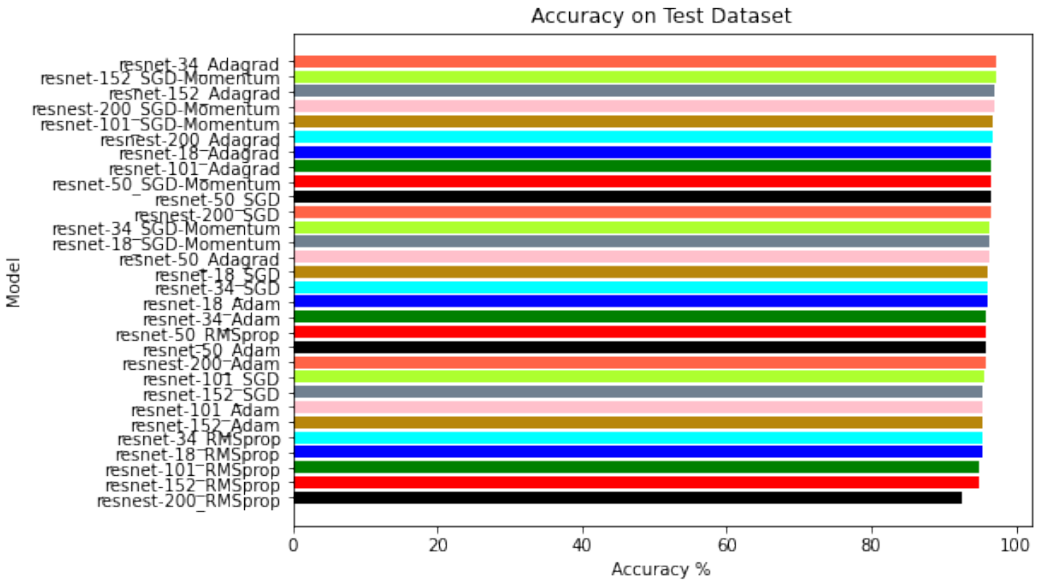


Figure 8. Test results of all the considered models with different optimizers.

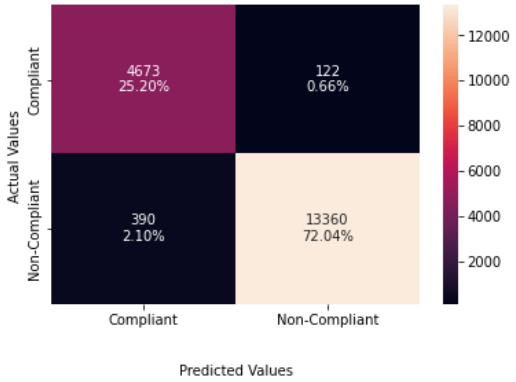


Figure 9. Confusion matrix of the results obtained from ResNet-34 with the Adagrad optimizer in the classification stage.

Since all the models performed with high accuracy, it is essential to know the training speed. The value of this is proportional to the number of layers. Therefore, the deeper model, i.e., ResNeSt-200, required a longer time for training, while the less deep model, i.e., ResNet-18, completed the training in less time. The time is proportionally related to the computational power needed to train the dataset, and powerful GPUs are required. Sometimes these computational resources are not available, and the experiments are difficult to perform on personal or desktop computers. From this phase, we can conclude that a light model such as ResNet-34 can accurately perform the COVID-19-mask-wearing classification task using Adagrad as an optimizer. Therefore, we can progress to implementing different CV techniques to obtain a better classification model.

4.2. Phase II: Improving the Dataset, Training the Classification Model for Two Classes, and Extending It to Three Classes

4.2.1. Datasets

The accuracy obtained for the previously described training was good (about 97.23%). In order to improve the accuracy of the model, it was necessary to check the dataset. We found that the dataset had many wrongly labeled images, and others were too noisy. For this reason, a data cleaning process was performed. An important tool for performing this activity was IBM Cloud Annotations. This is a simple web platform where datasets can be uploaded as zip files, classes added, and classification begun. The images are shown as a mosaic, and we could select them one by one or, by pressing the mouse's left button, move to the left and right, and up and down, selecting all the images we wished to place inside a class. In our particular project, the classes for the first dataset were Compliant and Non-Compliant, and for the second dataset they were Compliant, Incorrect, and Non-Compliant. After cleaning, we balanced the dataset, as shown in Table 4.

Table 4. New FMLD dataset after cleaning.

Dataset	**Classes**	
	Compliant	***Non-Compliant***
Training	20,937	20,937
Validation	1231	1231
Test	2463	2463
Total	24,631	24,631

In order to extend the classification task to three classes, it was necessary to look for another dataset similar to FMLD, because there were not enough images for the "incorrect" class. The dataset selected was the MMD, available on the Kaggle platform. Although we added images from another dataset, there was an imbalance in the number of images for

the incorrect class conmpared to the other two classes. Later, in the results subsection, we will discuss whether the imbalance affected the performance of the classification model. The new dataset formed by FMLD and MMD is detailed in Table 5.

Table 5. New dataset formed by FMLD + MMD datasets, extended to three classes.

Dataset	Classes		
	Compliant	**Incorrect**	**Non-Compliant**
Training	20,937	935	20,937
Validation	1231	55	1231
Test	2463	109	2463
Total	24,631	1099	24,631

In addition, data augmentation was performed using transforms available in the torchvision library. This type of data augmentation is different from the conventional type because it does not produce n images from one. Instead, before a batch of images enters the CNN, some of them are randomly modified by applying effects such as horizontal flips and rotation. This data augmentation helps to improve the variety of the dataset for training. Moreover, as the input to the ResNet is $3 \times 224 \times 224$, the images should be resized. Finally, normalization of the images using the means and standard deviations of the RGB channels of the ImageNet dataset is useful, in order to take full advantage of transfer learning, because the ResNet-18 pre-trained model was trained on the ImageNet dataset and our COVID-19 face mask images pixel values should be on a similar scale.

4.2.2. Training

The training was performed using light ResNet models such as ResNet-18 and ResNet-34, which are available in the torchvision repository. The decision to only test these two models was based on phase I, where it was proved that ResNet-34 was sufficient for performing the classification. Therefore, the objective was to improve the accuracy of ResNet-34 or try to obtain better accuracy with ResNet-18, which is a smaller model.

The cross-entropy loss function was used. The optimizer used for the training was Adagrad, with a learning rate of 0.001, a step size equal to one third of the number of epochs, and a gamma value equal to 0.1. The training consisted of 10 epochs for the model with two classes and 15 for the model with three classes. Finally, the batch size for training was 64, because we reached the maximum capacity of the GPU. Each epoch took approximately 1.14 s to complete.

4.2.3. Hardware

In order to perform the training, a Tesla P-100 GPU was used on a Google Colab instance with a 2.00 GHz Intel ® Xeon(R) CPU.

4.2.4. Results and Analysis

A test dataset was used to evaluate the model.

Two classes: The best accuracy obtained for the two classes (Compliant and Non-Compliant) was 99.6%, with ResNet-18. Additional metrics such as precision, recall, and F1-score are given in Table 6. In addition, a confusion matrix is plotted in Figure 10 to show the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). There were thirteen Compliant images classified as Non-Compliant and seven Non-Compliant images wrongly predicted as Compliant images. As we can see, the error rate is very low. Next, the wrongly predicted images are plotted according to each class.

Table 6. Additional evaluation metrics for COVID-19-mask-wearing classification model on new test dataset.

	Precision	Recall	F1-Score	Images
Compliant	1.00	0.99	1.00	2463
Non-Compliant	0.99	1.00	1.00	2463

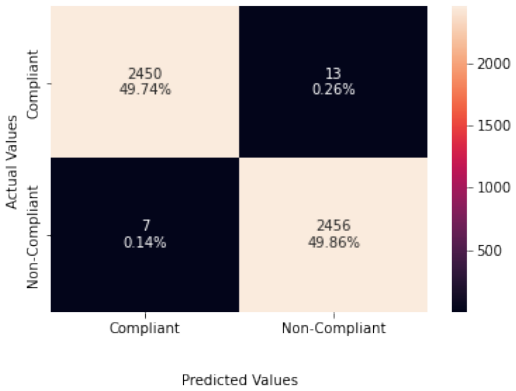


Figure 10. Confusion matrix for the testing of the ResNet-18 COVID-19-mask-wearing classification model on the new test dataset.

- **Compliant images classified as Non-Compliant**
  In Figure 11, there are three images (Figure 11a,h,i) with the wrong labels, i.e., the true label should be `Non-Compliant` but they were labeled as Compliant. The last six images were wrongly classified by the model. We can see the model has difficulties with the classification of images of industrial masks (Figure 11b,d) and with profile faces like the image in Figure 11c. We can use Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize the most representative parts of these images for the classification model.

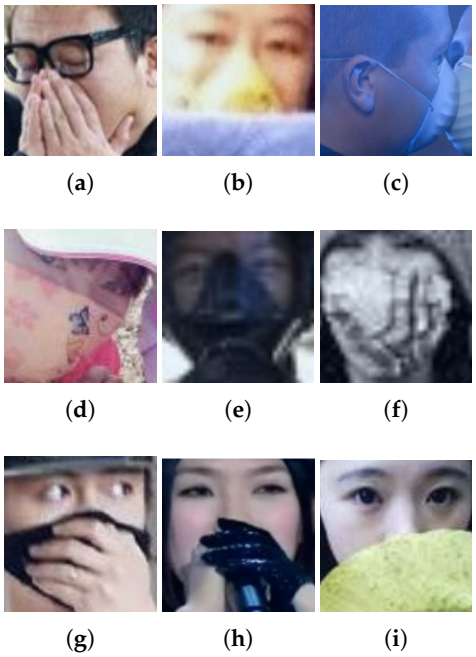


Figure 11. Compliant images classified as Non-Compliant.

Regarding Grad-CAMs, we find two principal problems. The first is related to the location of the most representative part of the image for the model, when this is not

around the nose tip as it should be (images in Figure 12a–d). The second is when the position of the most representative area of the image is correct but the model still cannot make a good classification (images in Figure 12e,f).

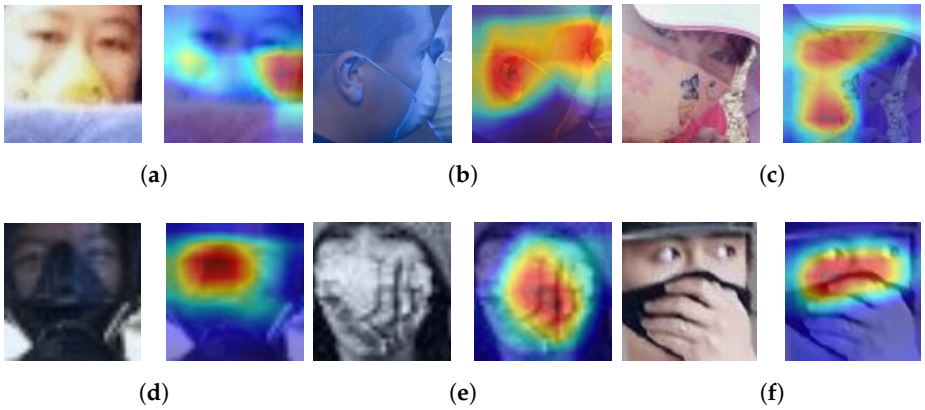


Figure 12. Compliant images classified as Non-Compliant, analyzed with Grad-CAMs.

- **Non-compliant images classified as Compliant**
  In Figure 13, the images are wrongly predicted, except the last one which was wrong labeled. The principal cause of the poor performance of the model for these images could be the presence of different objects around the noses. It is probable that the model regards these objects as a COVID-19 mask. The Grad-CAMs presented in Figure 14 help to prove this fact.
  For the images in Figure 14a,b, we note that the classification model is unable to distinguish between a COVID-19 mask and an object covering the nose. For the last two images (Figure 14c,d), the activation zone is wrong; therefore, these are poor predictions.

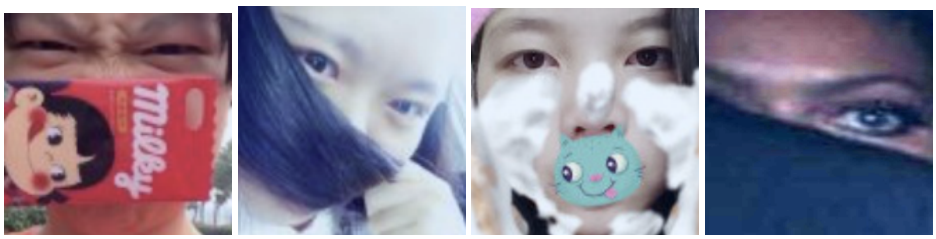


Figure 13. Non-compliant images classified as compliant.

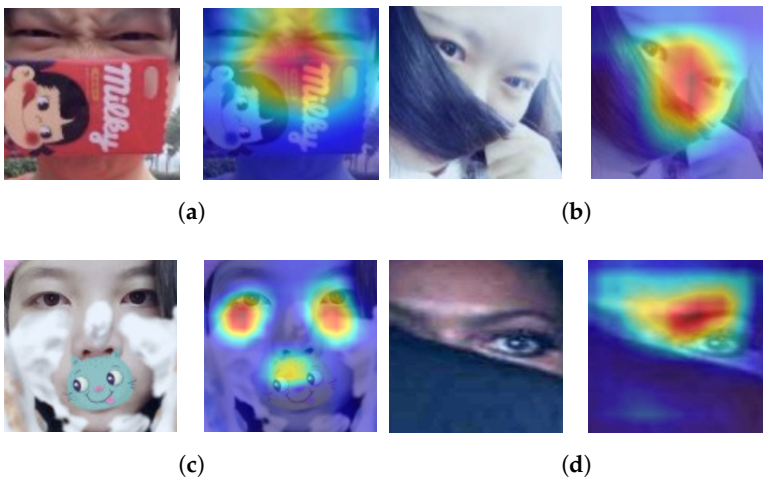


Figure 14. Non-Compliant images classified as Compliant, analyzed with Grad-CAMs.

In addition, we can see the results of the training for this model in the graph presented in Figure 15. The first plot represents the accuracy vs. the number of epochs for the training and validation datasets. Here, we can see that the accuracy on the training dataset increases with each epoch, but in the validation dataset, the best accuracy is in the second epoch, after which it decreases and finally remains constant. The second plot represents the loss vs. the number of epochs. Here we note that the loss decreases in the training dataset but not in the validation dataset. Next, the operation of the model with three classes will be explained.

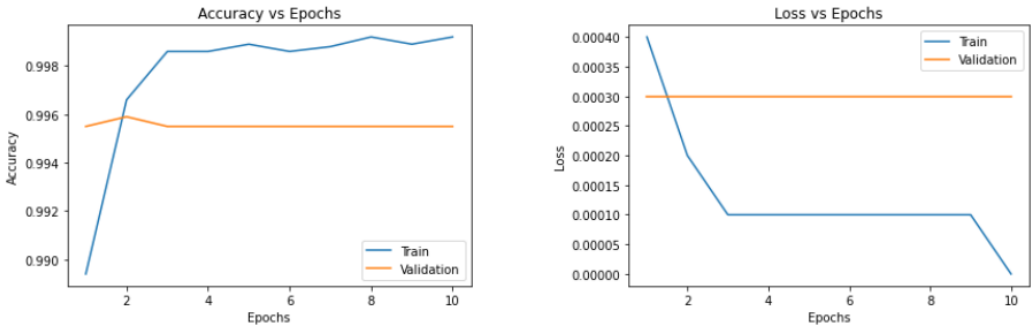


Figure 15. ResNet-18 accuracy and loss on training dataset for model with 2 classes.

Extension to three classes: The accuracy obtained for the model with three classes was 99.2%, with ResNet-18. Additional metrics are presented in Table 7, and a confusion matrix is given in Figure 16.

Table 7. Additional evaluation metrics for COVID-19-mask-wearing classification model on new test dataset for three classes.

	Precision	Recall	F1-Score	Images
Compliant	1.00	0.99	0.99	2463
Incorrect	0.86	0.90	0.88	109
Non-Compliant	0.99	1.00	0.99	2463

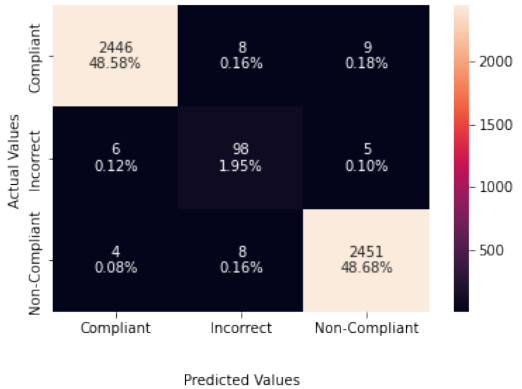


Figure 16. Confusion matrix for the testing of the ResNet-18 COVID-19-mask-wearing classification model on the new test dataset for three classes.

According to the confusion matrix in Figure 16, there were a few wrong predictions for each class. It is important to remember that there are only 109 images for the incorrect class; six were predicted as Compliant, and five were predicted as Non-Compliant. Eight images were predicted as Incorrect for the Compliant class, and nine were predicted as Compliant. Finally, four were predicted as Compliant for the Non-Compliant class, and eight were predicted as Incorrect. Therefore, we can conclude that the imbalance of the dataset in this particular case did not affect the accuracy of the model. However, the F1-score metric,

which considers how many errors the model has per class and their influence on the final performance, shows that the imbalance of the dataset affected the performance for the Incorrect class (see Table 7).

The confusion matrix shows summarized information about the model's results, but it would be helpful to know the images for which the prediction was wrong. For this reason, we plotted the wrongly predicted images per class, according to how they were classified in the labeling process.

- **Compliant images wrongly predicted**
  Figure 17 shows that most of the wrong predictions in this class were related to mistakes in the labeling of the data (images in Figure 17a–d,f,h,j,k,l,n,p), where the prediction was correct but the true label was wrong. In addition, we note that the model had difficulties with images of faces in profile (Figure 17h) and with faces that were very close (Figure 17e,i).

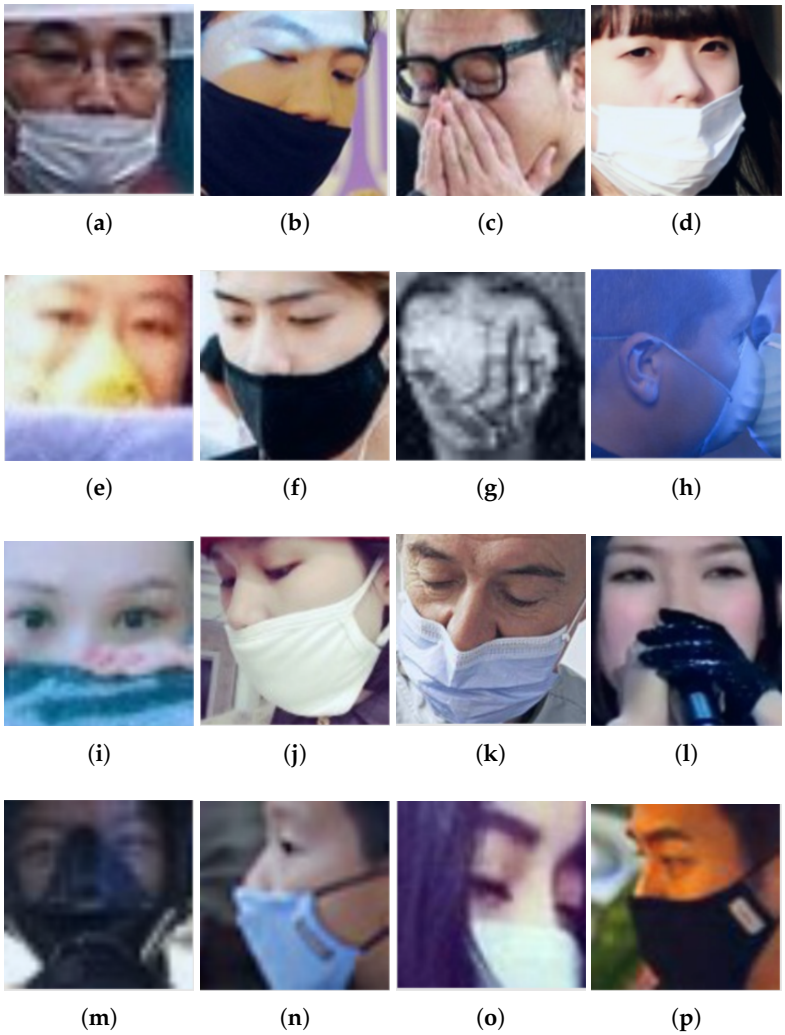
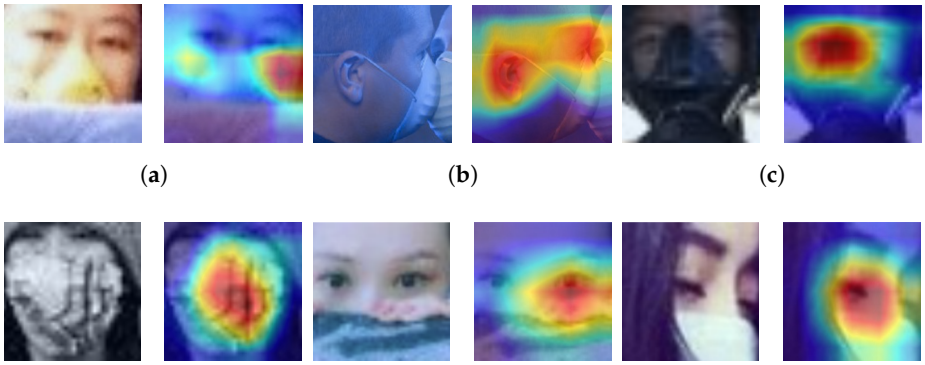


Figure 17. Compliant images wrongly classified. Each image has a predicted label assigned by the ResNet-18 classification model: (**a**) Incorrect; (**b**) Incorrect; (**c**) Non-Compliant; (**d**) Incorrect; (**e**) Non-Compliant; (**f**) Incorrect; (**g**) Non-Compliant; (**h**) Non-Compliant; (**i**) Non-Compliant; (**j**) Incorrect; (**k**) Incorrect; (**l**) Non-Compliant; (**m**) Non-Compliant; (**n**) Incorrect; (**o**) Non-Compliant; (**p**) Incorrect.

There is a way to understand better the reasons for wrong predictions using Grad-CAM. It enables us to visualize which parts of an image were more significant for a

model when making a prediction. The Grad-CAM was computed for the wrongly predicted images.

Grad-CAMs show a common factor: the most significant part of the image for the model for making the prediction is not the tip of the nose as it should be. In the case of Figure 18b, there is a clear problem related to faces in profile. This could be associated with the lack of profile images in the training dataset, because the proposed model focused on frontal faces. The subjects of the images in Figure 18a–c are wearing nose protection, but it is not a COVID-19 face mask; therefore, a poor prediction was made. Finally, the images in Figure 18e–f are close-ups of the face, and this may have caused the failure in the predicted label.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 18. Compliant images wrongly predicted, analyzed with Grad-CAMs.

- **Non-Compliant images wrongly predicted**

  One again, there were problems related to the wrong true labels in the images in Figure 19e–h. In addition, there are wrong predictions, especially based on the confusion of COVID-19 face masks with other objects (images in Figure 19a–d). This fact can be seen with the use of Grad-CAMs in Figure 20.

(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 19. Non-Compliant images wrongly classified. Each image has a predicted label assigned by the ResNet-18 classification model: (**a**) Compliant; (**b**) Incorrect; (**c**) Compliant; (**d**) Incorrect; (**e**) Incorrect; (**f**) Incorrect; (**g**) Incorrect; (**h**) Incorrect.

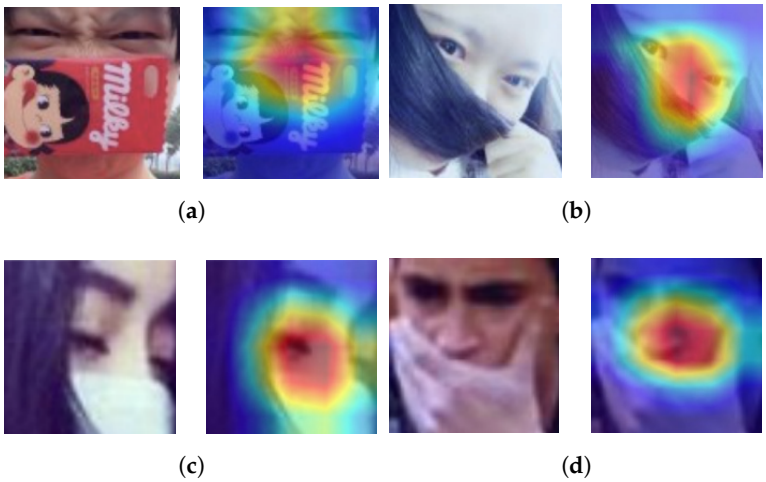


Figure 20. Non-Compliant images wrongly predicted, analyzed with Grad-CAMs.

According to the Grad-CAM, the more significant parts of the image are located around the nose. Here, we can see objects covering noses that are different from COVID-19 face masks. Therefore, it is probable that the model can sometimes become confused when a nose is covered with some object other than a mask. To solve this problem, it can be helpful to put images with noses covered by other objects and labeled as "Non-Compliant" in the training datasets, to allow the model learn this new feature.

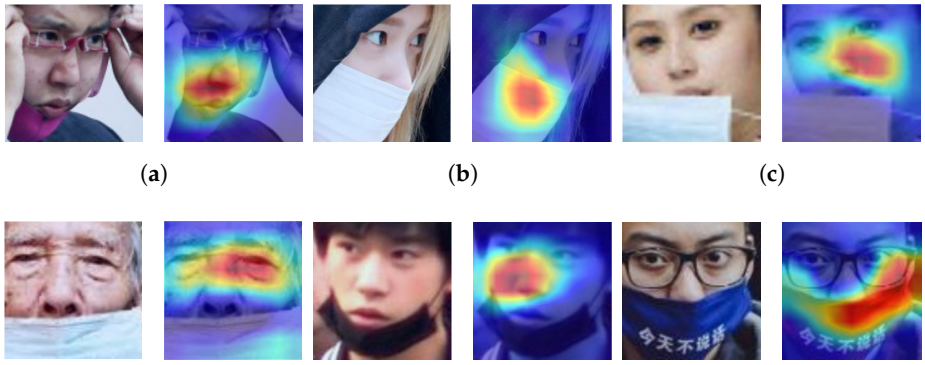
- **Incorrect images wrongly predicted**
  The class "Incorrect" is a particular case, because it can be relative, according to the position of the COVID-19 face mask with respect to the nose. For example, some people consider that the mask is being used incorrectly if it only covers the nasal base. Other people consider the use of the COVID-19 face mask to be incorrect if it does not cover the ridge of the nose entirely. Finally, since many people prefer to put the COVID-19 face mask under the chin to avoid taking it off completely when they want to do not wear it for a short period of time, the position of the COVID-19 face mask under the chin can be considered as "Incorrect". As we can see, we have different concepts for the Incorrect class, and for this reason, there are some wrong predictions (see Figure 21) related to this fact, which Grad-CAMs can help us to describe.
  According to Grad-CAMs, the classification model focuses on two principal parts of the image. The first one is around the nasal base, as in the images in Figure 22a,c,e. The second is related to the identification of the COVID-19 face mask (images in Figure 22b,f). In the first case, it may be impossible to classify the image as Incorrect because the model does not focus on the COVID-19 mask, as it is under the chin. As the model identifies the absence of the COVID-19 mask around the nose, the predicted label is Non-Compliant. For the second case, the model only pays attention to the presence of the COVID-19 mask. For this reason, the model cannot identify that the mask is only covering the nasal base, i.e., the mask is used incorrectly, but the model predicts it as Compliant.

Similarly to the previous subsection for the two models, we can see the results of the training for this model in Figure 23. In the first plot, we can see that the accuracy of the training dataset increases. In the final epochs, it converges, but in the validation dataset the accuracy increases and decreases, reaching a maximum value in epoch number 14. On the other hand, the second plot shows that in the training dataset, the loss decreases to 0.0001 within the first four epochs and then remains constant up to the final epoch. In contrast, the loss decreases as the epochs increase in the validation dataset, with a minimum loss of 0.004.

(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**)

Figure 21. Incorrect images wrongly classified. Each image has a predicted label assigned by the ResNet-18 classification model: (**a**) Non-Compliant; (**b**) Compliant; (**c**) Non-Compliant; (**d**) Non-Compliant; (**e**) Non-Compliant; (**f**) Compliant; (**g**) Compliant.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 22. Incorrect images wrongly predicted, analyzed with Grad-CAMs.

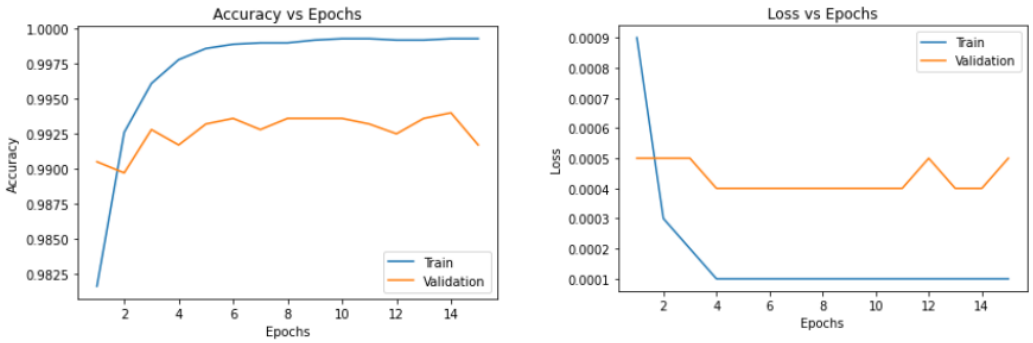


Figure 23. ResNet-18 accuracy and loss on the training dataset for the model with three classes.

4.3. Visualizing Some Feature Maps and Filters

Next, the feature maps are shown. As mentioned previously, the feature maps are images/outputs of a layer after applying a group of filters, where the output becomes the new input for the next layer until the final layer is reached. Features maps are helpful for

understanding deep neural networks better, because we can see the features the model pays attention to and what filters are applied. The feature maps for each of the seventeen convolutional layers, after applying filters, are shown in Figure 24.

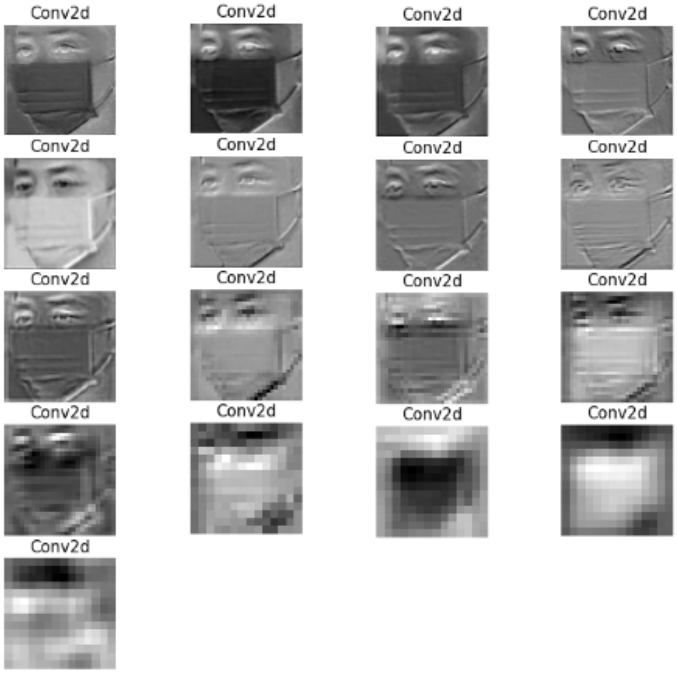


Figure 24. Feature maps corresponding to each convolutional layer of ResNet-18.

ResNet-18 Filters

The values used to make the filter plots are the weights of the kernels. For example, the kernel size for the first convolutional layer is 7×7 (64 filters in total), and the filter looks as shown in Figure 25. As the image passes through the filters, it undergoes transformations. In Figure 26, there are five images passing through seven filters. The filters determine which parts of the image the model will focus on. Plotting the feature maps and filters in grayscale makes it easy to understand what is happening in a convolution. Therefore, we have plotted all the 64 filters of the first convolutional layer, together with the feature maps they produced. This result is shown in Figure 27.

Considering one of the feature maps, we see that some parts of it are dark and others are bright. This is due to the dark and bright parts of the filters. The reason for a part of the filter being dark or bright is the numerical value of the pixel. We know that a pixel is composed of three values in RGB images, from 0 to 255 (values near zero are black and values near 255 are white). The values for the pixels are the weights. Low weights mean dark pixels, and high values mean bright pixels. Finally, the model will pay more attention to the parts of the image where the element-wise product of the weight and the pixel value is high. This means that the bright parts of the image are responsible for activating a particular layer's neurons, depending on the values of its weights. This can be interpreted as what the neural network sees. In this particular case, to classify the position of the COVID-19 face mask, we can see that some filters focus on the background of the image while others focus on the person wearing the mask. In some cases, the background is dark and the person becomes bright or vice versa. In addition, some filters produce an outline of the mask.

Finally, we see the evolution of an image over the first 64 filters of the convolutional layers 1, 5, 10, 15, and 17, with their respective feature maps, in Figure 28. An important fact to consider is that the number of kernels of a convolutional layer can be greater than 64, but plotting all the kernels can consume a great deal of space. Therefore, we plot the 64 filters of each convolutional layer for demonstration purposes.

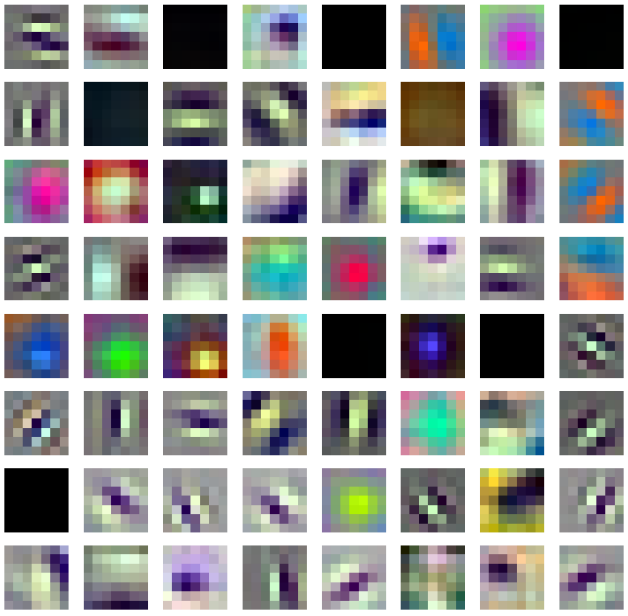


Figure 25. 64 Filters corresponding to the first convolutional layer of ResNet-18.

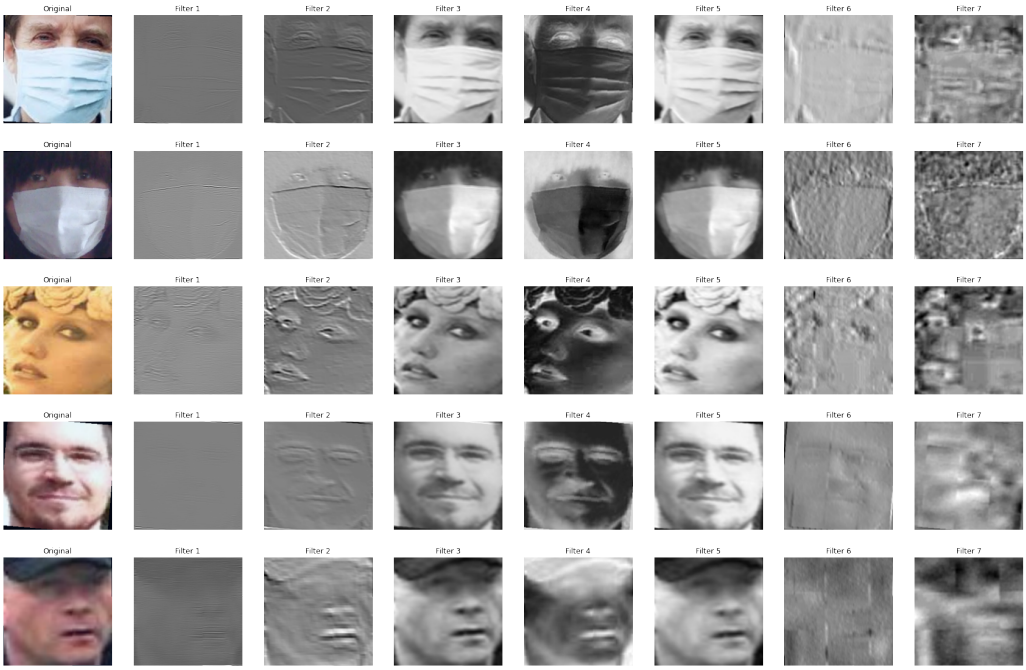
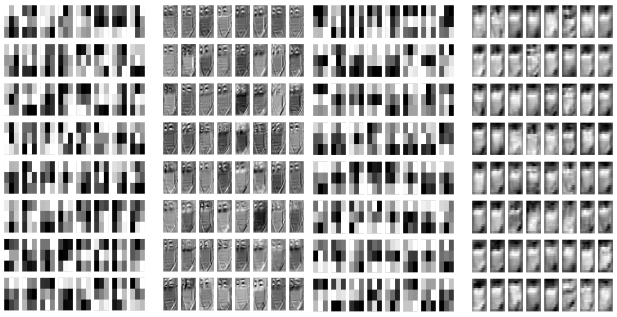


Figure 26. Five images together with their feature maps after applying seven different filters in the first convolutional layer.

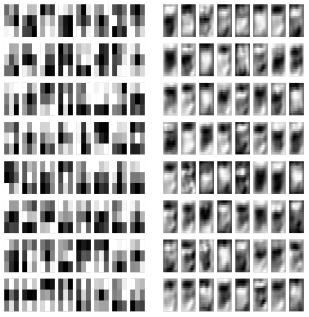
It is clear to see that, going deeper into the convolutional layers, the patterns on the features maps become more challenging to see for the human eye, because the details of the image disappear. However, the neural network can identify the pattern to make the classification. The last feature maps look noisy, but they are the most important for the fully connected neurons (classification layer).

Figure 27. All 64 filters of the first convolutional layer with the feature maps produced.

(**a**) (**b**)



(**c**) (**d**)



(**e**)

Figure 28. Filters with their respective feature maps for convolutional layers 1, 5, 10, 15 and 17: (**a**) convolutional layer 1; (**b**) convolutional layer 5; (**c**) convolutional layer 10; (**d**) convolutional layer 15; (**e**) convolutional layer 17.

4.4. Grad-CAMs through the Layers

It is possible to see the Grad-CAMs' evolution through each layer. We recall that ResNet-18 has 18 layers. However, the term "layer" has an additional meaning: a compound of two basic blocks. Therefore, taking this latter meaning, ResNet-18 has four layers, each one composed of two basic blocks. In the same way, a basic block is formed by two operations, and an operation is formed by a convolution, a batch normalization, and a ReLU activation, except for the last operation of a basic block. Therefore, we can appreciate the activations of each basic block after applying the operations over an image of each class. In this way, we can appreciate that the beginning layer identifies general features and then, as we go deeper, the activations are focused on the COVID-19 face mask and the area around the nose (see Figure 29).

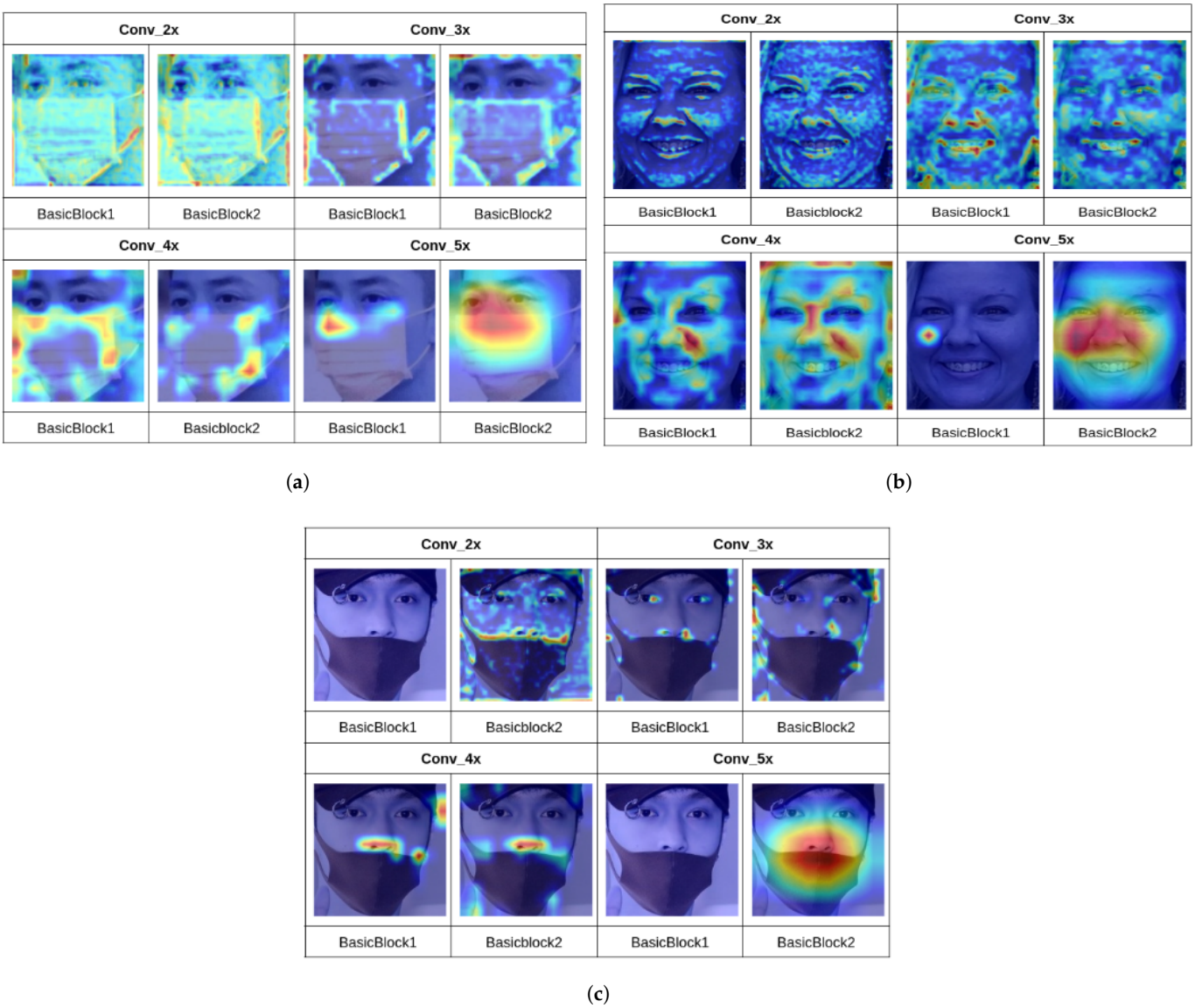
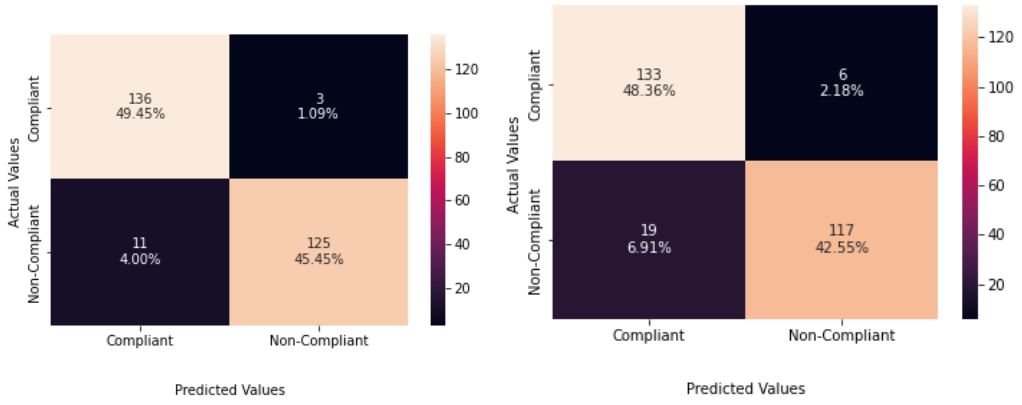


Figure 29. Grad-CAMs per convolutional layer for an image in each class: (**a**) Compliant; (**b**) Non-Compliant; (**c**) Incorrect.

COVID-19 Mask Classification Model with ResNet-18 against Other Approaches

In [49], the authors use two different neural network combinations. First, they use MobileNetV2 as a feature extractor and support vector machine (SVM) to make the classification. In the second experiment, they use VGG-19 as a feature extractor and *k*-nearest

neighbours (KNN) for the classification stage. The confusion matrices for both approaches are shown in Figure 30.



(**a**) (**b**)

Figure 30. Confusion matrices corresponding to MobileNetV2 + SVM and VGG19 + KNN: (**a**) MobileNetV2 and SVM; (**b**) VGG19 and KNN.

To enable a fair comparison, we used the same dataset with the same number of images to test the model (139 images for the Compliant class and 136 images for the Non-Compliant class). After performing the classification with our model, the confusion matrix presented in Figure 31 was obtained. If we compare the three confusion matrices, it is clear that our model results in better classification, especially for the Non-Compliant class, where our model reduced the number of wrong predictions from 11 (MobileNetV2 + SVM, Figure 30a) and 19 (VGG19 + KNN, Figure 30b) to only 3. In addition, we can use other metrics to see the results.

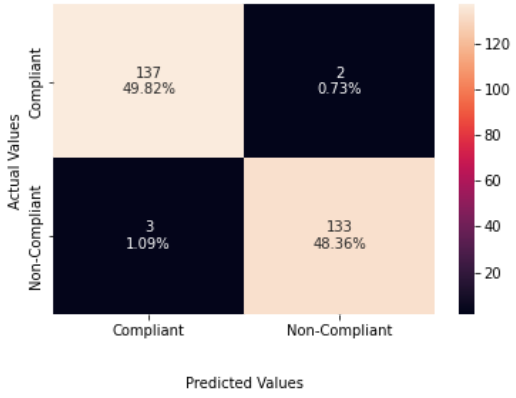


Figure 31. ResNet-18 results after being applied on the other dataset.

As we can see from Table 8, ResNet-18 has a recall and precision of 98%, which is an improvement of around 3% over the better model presented in [49], i.e., the MobileNetV2 with SVM.

Table 8. ResNet-18 in comparisson with other approaches.

Models	Recall	Precision
MobileNetV2 and SVM	94.84%	95.08%
VGG19 and KNN	90.09%	91.3%
ResNet-18	98%	98%

5. Conclusions

5.1. General Conclusions

In this work, we presented a CNN capable of classifying face mask images for use during the COVID-19 pandemic. Our model classifies the use of face masks into three different classes: Compliant (when people are wearing the mask properly), Incorrect (when the person is not wearing the mask according to the WHO guidelines, i.e., the mask does not completely cover the mouth and nose), and Non-Compliant (when people are wearing no COVID-19 face mask). We began by reviewing bibliographic material on different approaches to the problem of the classification of face-mask-wearing images, from basic methods to those that are considered to be state of the art. Finally, we achieved a solid two-stage pipeline capable of performing the task assigned to this project. The metric results showed that pipeline had excellent performance, encouraging its use in applications that require face detection, followed by a face mask classification stage.

Firstly, we tested DL models, especially those in the ResNet family, using different optimizers, in order to determine the performance of each one in the task of classifying the COVID-19 mask wearing into two classes: Compliant and Non-Compliant. As a result of this preliminary study, our data showed that all models had an accuracy of over 94% (except ResNeSt-200 with RMSProp, which had an accuracy of 92%). ResNet-34 with Adagrad was selected as the best classification model, since the accuracy was 97.24% and it required less time to train, as it only has 34 layers compared with deeper models such as ResNet-50 or ResNet-101.

The ResNet-34 model and its accuracy of 97.24% were taken as the starting point for the second part of the work. In this part, we improved the general performance and accuracy by applying different computer vision techniques, both in the pre-processing of the dataset and in the training of the classification model. The most important techniques included cleaning, relabeling, and balancing the dataset, with the use of torchvision transforms to perform data augmentation on each training batch, guaranteeing the variability of the dataset. All these improvements led to better results in the COVID-19-mask-wearing classification model in terms of accuracy, time to train, and the model's size, even when the number of classes was extended from two (Compliant and Non-Compliant) to three (Compliant, Incorrect, and Non-Compliant). The best model was ResNet-18 in both cases, with 99.6% accuracy for two classes and 99.2% accuracy for three classes, using Adagrad as an optimizer as in the preliminary study. In addition, the setting of the hyperparameters was described in Section 4, for the models with both two and three classes.

Furthermore, as an excellent model was obtained, in order to try to decipher what many people call a black box, we used Grad-CAMs to discover which features or parts of the image the model paid attention to when giving a prediction for a specific class. Consequently, we discovered that the model focused on the zone around the nose, especially the tip of the nose, to see the presence or absence of the COVID-19 mask and to give a prediction. Additionally, we presented many feature maps and the associated filters, i.e., the kernels which produce them across the model layers, to observe the evolution of an image from the shallow layers up to the last convolutional layer, which is the one before the fully connected layer, i.e., the classification layer.

5.2. Future Work

Future work could be grounded on the deployment of our model in embedded systems. It could be helpful to transform our Pytorch COVID-19-mask-wearing classification model using NVIDIA TensorRT, to make the transition to production using NVIDIA DeepStream SDK. TensorRT allows us optimize our Pytorch model through techniques such as punning, layer/tensor fusion, kernel auto-tuning, etc., while NVIDIA DeepStream SDK provides us with a multi-platform approach to deploying Vision AI applications and services in the real world. An important task that needs to be addressed is increasing the number of images available for the Incorrect class dataset, to extend the identification of the position of the mask when people are not protecting their nose or are wearing the mask under the

chin. Finally, in terms of datasets, it would be helpful to try to conduct the training with a lower number of images per class, to probe whethr it is possible to obtain higher or similar accuracy to the presented ResNet-18 model with a dataset of smaller size, given that, in this work, we showed that the model was capable of identifying the Incorrect class accurately with only 1100 images.

Author Contributions: F.C.: conceptualization, methodology, software, formal analysis, investigation, writing—original draft preparation, visualization, and resources; A.C.: investigation, software, and visualization; L.M.S.-M.: formal analysis, writing—review, and funding acquisition; D.H.P.-O.: validation, writing—review, methodology, supervision, and funding acquisition; M.E.M.-C.: interpretation of data, writing—review, validation, and project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by University of Cauca.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code for the study presented in this article and the access link to the datasets used are available on https://github.com/FabricioCrespo/covid_mask_detection.git (accessed on 9 May 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Velavan, T.P.; Meyer, C.G. The COVID-19 epidemic. *Trop. Med. Int. Health* **2020**, *25*, 278. [CrossRef] [PubMed]
2. Yuki, K.; Fujiogi, M.; Koutsogiannaki, S. COVID-19 pathophysiology: A review. *Clin. Immunol.* **2020**, *215*, 108427. [CrossRef]
3. Elachola, H.; Ebrahim, S.H.; Gozzer, E. COVID-19: Facemask use prevalence in international airports in Asia, Europe and the Americas, March 2020. *Travel Med. Infect. Dis.* **2020**, *35*, 101637. [CrossRef]
4. Lee, B.Y. How Israel Ended Outdoor Face Mask Mandates with the Help of Covid-19 Vaccines. 2021. Available online: https://www.forbes.com/sites/brucelee/2021/04/20/how-israel-ended-outdoor-face-mask-mandates-with-the-help-of-covid-19-vaccines/?sh=63598b46680e (accessed on 12 April 2022).
5. Feng, S.; Shen, C.; Xia, N.; Song, W.; Fan, M.; Cowling, B.J. Rational use of face masks in the COVID-19 pandemic. *Lancet Respir. Med.* **2020**, *8*, 434–436. [CrossRef]
6. Howard, J.; Huang, A.; Li, Z.; Tufekci, Z.; Zdimal, V.; van der Westhuizen, H.M.; von Delft, A.; Price, A.; Fridman, L.; Tang, L.H.; et al. An evidence review of face masks against COVID-19. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2014564118. [CrossRef]
7. Eikenberry, S.E.; Mancuso, M.; Iboi, E.; Phan, T.; Eikenberry, K.; Kuang, Y.; Kostelich, E.; Gumel, A.B. To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic. *Infect. Dis. Model.* **2020**, *5*, 293–308. [CrossRef]
8. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustain. Cities Soc.* **2021**, *65*, 102600. [CrossRef]
9. Castro, R.; Pineda, I.; Lim, W.; Morocho-Cayamcela, M.E. Deep Learning Approaches Based on Transformer Architectures for Image Captioning Tasks. *IEEE Access* **2022**, *10*, 33679–33694. http://doi.org/10.1109/ACCESS.2022.3161428. [CrossRef]
10. Castro, R.; Pineda, I.; Morocho-Cayamcela, M.E. *Hyperparameter Tuning over an Attention Model for Image Captioning, Proceedings of the Information and Communication Technologies, Ann Arbor, MI, USA, 3–6 June 2021*; Salgado Guerrero, J.P., Chicaiza Espinosa, J., Cerrada Lozada, M., Berrezueta-Guzman, S., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 172–183.
11. Morocho-Cayamcela, M.E.; Lim, W. Pattern recognition of soldier uniforms with dilated convolutions and a modified encoder-decoder neural network architecture. *Appl. Artif. Intell.* **2021**, *35*, 476–487. http://doi.org/10.1080/08839514.2021.1902124. [CrossRef]
12. Moreno-Revelo, M.Y.; Guachi-Guachi, L.; Gómez-Mendoza, J.B.; Revelo-Fuelagán, J.; Peluffo-Ordóñez, D.H. Enhanced Convolutional-Neural-Network Architecture for Crop Classification. *Appl. Sci.* **2021**, *11*, 4292. http://doi.org/10.3390/app11094292. [CrossRef]
13. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 9396–9405. http://doi.org/10.1109/CVPR.2019.00963. [CrossRef]
14. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* **2021**, *167*, 108288. [CrossRef] [PubMed]
15. Qin, B.; Li, D. Identifying Facemask-Wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19. *Sensors* **2020**, *20*, 5236. [CrossRef] [PubMed]

16. Anaya-Isaza, A.; Mera-Jiménez, L.; Cabrera-Chavarro, J.M.; Guachi-Guachi, L.; Peluffo-Ordóñez, D.; Rios-Patiño, J.I. Comparison of Current Deep Convolutional Neural Networks for the Segmentation of Breast Masses in Mammograms. *IEEE Access* **2021**, *9*, 152206–152225. http://doi.org/10.1109/ACCESS.2021.3127862. [CrossRef]
17. Batagelj, B.; Peer, P.; Štruc, V.; Dobrišek, S. How to Correctly Detect Face-Masks for COVID-19 from Visual Information? *Appl. Sci.* **2021**, *11*, 2070. [CrossRef]
18. Loop Hit. Medical Mask Dataset: Humans in the Loop. 2022. Available online: https://humansintheloop.org/medical-mask-dataset (accessed on 1 June 2022).
19. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. Wider face: A face detection benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5525–5533.
20. Jain, V.; Learned-Miller, E. *Fddb: A Benchmark for Face Detection in Unconstrained Settings;* UMass Amherst Technical Report; UMass: Amherts, MA, USA, 2010.
21. Zhu, X.; Ramanan, D. Face detection, pose estimation, and landmark localization in the wild. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2879–2886.
22. Jain, A.K.; Klare, B.; Park, U. Face recognition: Some challenges in forensics. In Proceedings of the 2011 IEEE International Conference on Automatic Face Gesture Recognition (FG), Santa Barbara, CA, USA, 21–23 March 2011; pp. 726–733. http://doi.org/10.1109/FG.2011.5771338. [CrossRef]
23. Jain, A.K.; Klare, B.; Park, U. Face Matching and Retrieval in Forensics Applications. *IEEE MultiMedia* **2012**, *19*, 20. http://doi.org/10.1109/MMUL.2012.4. [CrossRef]
24. Zeng, J.; Zeng, J.; Qiu, X. Deep learning based forensic face verification in videos. In Proceedings of the 2017 International Conference on Progress in Informatics and Computing (PIC), Nanjing, China, 27–29 October 2017; pp. 77–80. http://doi.org/10.1109/PIC.2017.8359518. [CrossRef]
25. Ge, S.; Li, J.; Ye, Q.; Luo, Z. Detecting masked faces in the wild with lle-cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2682–2690.
26. Cabani, A.; Hammoudi, K.; Benhabiles, H.; Melkemi, M. MaskedFace-Net–A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health* **2021**, *19*, 100144. [CrossRef]
27. Wang, Z.; Wang, G.; Huang, B.; Xiong, Z.; Hong, Q.; Wu, H.; Yi, P.; Jiang, K.; Wang, N.; Pei, Y.; et al. Masked face recognition dataset and application. *arXiv* **2020**, arXiv:2003.09093.
28. Roy, B.; Nandy, S.; Ghosh, D.; Dutta, D.; Biswas, P.; Das, T. MOXA: A deep learning based unmanned approach for real-time monitoring of people wearing medical masks. *Trans. Indian Natl. Acad. Eng.* **2020**, *5*, 509–518. [CrossRef]
29. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition—CVPR, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. I.
30. Li, H.; Lin, Z.; Shen, X.; Brandt, J.; Hua, G. A convolutional neural network cascade for face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5325–5334.
31. Farfade, S.S.; Saberian, M.J.; Li, L.J. Multi-view face detection using deep convolutional neural networks. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, 23–26 June 2015; pp. 643–650.
32. Sun, X.; Wu, P.; Hoi, S.C. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing* **2018**, *299*, 42–50. [CrossRef]
33. Zhang, S.; Chi, C.; Lei, Z.; Li, S.Z. Refineface: Refinement neural network for high performance face detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4008–4020. [CrossRef]
34. Gao, J.; Yang, T. Face detection algorithm based on improved TinyYOLOv3 and attention mechanism. *Comput. Commun.* **2022**, *181*, 329–337. [CrossRef]
35. Li, Y. Face Detection Algorithm Based on Double-Channel CNN with Occlusion Perceptron. *Comput. Intell. Neurosci.* **2022**, *2022*, 3705581 . [CrossRef] [PubMed]
36. Wu, P.; Li, H.; Zeng, N.; Li, F. FMD-Yolo: An efficient face mask detection method for COVID-19 prevention and control in public. *Image Vis. Comput.* **2022**, *117*, 104341. [CrossRef] [PubMed]
37. Zhang, J.; Han, F.; Chun, Y.; Chen, W. A novel detection framework about conditions of wearing face mask for helping control the spread of covid-19. *IEEE Access* **2021**, *9*, 42975–42984. [CrossRef]
38. Lin, H.; Tse, R.; Tang, S.K.; Chen, Y.; Ke, W.; Pau, G. Near-realtime face mask wearing recognition based on deep learning. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–7.
39. Xiong, Y.; Zhu, K.; Lin, D.; Tang, X. Recognize complex events from static images by fusing deep channels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015, pp. 1600–1609.
40. Deng, J.; Guo, J.; Zhou, Y.; Yu, J.; Kotsia, I.; Zafeiriou, S. Retinaface: Single-stage dense face localisation in the wild. *arXiv* **2019**, arXiv:1905.00641.
41. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

43. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. Resnest: Split-attention networks. *arXiv* **2020**, arXiv:2004.08955.
44. Morocho-Cayamcela, M.E.; Lee, H.; Lim, W. Machine Learning to Improve Multi-Hop Searching and Extended Wireless Reachability in V2X. *IEEE Commun. Lett.* **2020**, *24*, 1477–1481. http://doi.org/10.1109/LCOMM.2020.2982887. [CrossRef]
45. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
46. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
47. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [CrossRef]
48. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
49. Oumina, A.; El Makhfi, N.; Hamdi, M. Control the covid-19 pandemic: Face mask detection using transfer learning. In Proceedings of the 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, 2–3 December 2020; pp. 1–5.