Title of the article: Transfer Learning for Mobile Real-Time Face Mask Detection and Localisation

Full name, postal address, e-mail and telephone number of the corresponding author.: Francesco Mercaldo, Via Francesco De Sanctis 1, Campobasso, Italy, francesco.mercaldo@unimol.it, tel. +39 0874 40 41

Full name, department, institution, city and country of all co-authors. Francesco Mercaldo, Department of Medicine and Health Sciences “Vincenzo Tiberio”, University of Molise, Campobasso, Italy and Antonella Santone, Department of Medicine and Health Sciences “Vincenzo Tiberio”, University of Molise, Campobasso, Italy

Up to five keywords or phrases suitable for use in an index: face mask, deep learning, artificial intelligence

Word count, excluding title page, abstract, references, figures and tables: 3300
Transfer Learning for Mobile Real-Time Face Mask Detection and Localisation

Francesco Mercaldo\textsuperscript{1} and Antonella Santone\textsuperscript{1}

\textsuperscript{1}Department of Medicine and Health Sciences "Vincenzo Tiberio", University of Molise, Campobasso, Italy

Abstract

Objective: Due to the COVID-19 pandemic, our daily habits have suddenly changed. Gatherings are forbidden and even when it is possible to leave the home, for health or work reasons, it is necessary to wear a face mask to reduce the possibility of contagion. In this context, it is crucial to detect violations by people who do not wear a face mask.

Materials and Methods: For these reasons, in this paper we introduce a method aimed to automatically detect whether people are wearing a face mask. We design a transfer learning approach by exploiting the MobileNetV2 model to identify in images/video streams face mask violations. Moreover the proposed approach is able to localise the area related to the face mask detection with the relative probability.

Results: To assess the effectiveness of the proposed approach, we evaluate a dataset composed by 4095 images related to people wearing and not wearing a face mask, obtaining an accuracy of 0.98 in face mask detection.

Discussion and Conclusion: The experimental analysis shows that the proposed method can be successfully exploited for the face mask violations detection. Moreover we highlight that it is working also on device with limited computational capability and it is able to process in real time images and video streams, making our proposal able to be applied in the real-world.

1 Introduction

The severe acute respiratory syndrome Coronavirus-2 (SARS-CoV-2) is the name given to the new coronavirus discovered in 2019 [1]. COVID-19 is the name given to the disease associated with this new kind of virus. SARS-CoV-2 is a new coronavirus strain that has not previously been identified in humans.
Some coronaviruses can be transmitted from person to person, usually after close contact with an infected patient, such as between family members or in a healthcare setting [2].

The new coronavirus, responsible for the COVID-19 respiratory disease, can also be transmitted from person to person, through close contact with a probable or confirmed case.

Current evidence suggests that SARS-CoV-2 spreads from person to person:

- directly;
- indirectly (through contaminated objects or surfaces);
- by close contact with infected persons through secretions from the mouth and nose (saliva, respiratory secretions or droplets).

When a sick person coughs, sneezes, talks or sings, these secretions are released from the mouth or nose. People who are in close contact (less than 1 meter) with an infected person can become infected if the droplets enter the mouth, nose or eyes [3].

Preventive measures are therefore to maintain a physical distance of at least one meter, wash your hands frequently and wear a mask.

Sick people can release infected droplets on objects and surfaces (called fomites) when they sneeze, cough, or touch surfaces (tables, handles, handrails). By touching these objects or surfaces, other people can become infected by touching their eyes, nose or mouth with contaminated (not yet washed) hands [4].

This is why it is essential to wash hands [5] properly and regularly with soap and water or an alcohol-based product and to clean surfaces frequently [6].

Moreover, to avoid the spread of the pandemic, it is mandatory for people to always wear face masks [7]. These must be worn in indoor places other than private houses and also in all outdoor places, except in cases where, due to the characteristics of the place or the factual circumstances, the condition of isolation is continuously guaranteed [8].

With the aim to ensure the safety of the people and places we daily frequent in the COVID-19 pandemic, in this paper we present an approach aimed to automatically detect whether people are wearing or not a face mask.

The main aim of the proposed approach is the face mask real-time detection (from both video and/or image streams). To this purpose, we exploit deep learning techniques, in particular transfer learning is considered in this paper to build an accurate model aimed to detect people not wearing a face mask (even if there are more people in the image). Moreover, the proposed method is aimed also to localise people within the image and/or video stream, also associating a detection accuracy percentage for each individual person detected.

Below the distinctive points of the proposed approach:

- we propose an approach aimed to automatically and silently detect whether people are wearing the face mask;
• a way to understand the reason why the classifier outputs a certain detection making the proposed method explainable i.e., by automatically drawing the area interested by the detection (thus providing to the analyst the area of the image that brought the model to output a certain prediction);

• we resort to transfer learning, in detail we based the proposed architecture on the top of the MobileNet V2 network, widespread to efficiently work on device with limited resources (as, for instance, smartphones, tablets, Google Coral and Raspberry Pi devices;

• it is completely real-time i.e., is working on both images and live video streams (and for this reason it can be implemented in devices such as surveillance cameras);

• we evaluated an extended dataset composed by a total of 4095 images (2165 related to people wearing a face mask and 1930 of people not wearing a face mask);

• an accuracy equal to 0.98 is obtained.

The paper continues in the following way: in the next section we present the proposed approach for mobile real-time face mask detection, in Section 3 we present the study we conducted to assess the effectiveness of the propose method; in Section 4 we present the state-of-the-art literature in the face mask detection context and, finally, in the last section conclusion and future work are discussed.

2 The Face Mask Detection Method

In this section we present the proposed method for real-time face mask detection. Our approach is based on transfer learning i.e., an artificial intelligence technique that serves to adapt an artificial intelligence to a task other than the one for which it was initially trained. The fundamental knowledge learned by an artificial intelligence in a given domain can be directly reapplied to another domain by simply "retuning", thus avoiding retraining it from scratch.

In this paper we experiment with the MobileNetV2 [9] model i.e., a deep convolutional neural network composed by 53 layers. To train this network researchers considered the ImageNet [10] database composed by more than one million images. It is able to classify images into 1000 object categories as, for instance, keyboard, mouse, pencil and many animals. As a result, the network has learned feature-rich representations for a wide range of images. The network has an image input size of 224x224.

In Figure 1 the work ow of the proposed approach is depicted.

As shown from Figure 1 the method we propose relies in two main steps: Training, aimed to generated a model, and Validation, aimed to evaluate the model obtained in the previous step.
Figure 1: The workflow of proposed approach.

For model generation we rstly consider a Face Mask dataset, composed by several images representing people wearing a face mask and people not wearing a face mask. We manually inspected the images belonging to the dataset in order to accurately annotate each image (with the mask or with the no mask label). This represents an important task, considering that the model needs a labelled dataset and that labelling errors lead to a noise in the dataset which is reflected in a decay of the performance of the proposed approach.

Once obtained a dataset composed by an adequate amount of data, in the Model Building we consider a deep learning network designed by authors, exploiting the architecture proposed by the MobileNetV2 network of Google. MobileNet is a model designed to be performed primarily on mobile and low capability devices (for instance, Raspberry Pi) so as to ensure portability and speed of execution at the expense, however, of the general accuracy in phase of detection [11]. Basically, this is a neural network aimed at image classification but, with the application of the Single Shot Multibox Detector (SSD) detector, it was converted to the object Detection task. The architecture is this network is based on the one of the VGG-16 network [12], but removing the layers fully connected; the reasons why this network was used as a basis are its performances in high image classification quality and its popularity in the problems in which the transfer Learning technique helps improve results [13]. Instead of fully connected layers, a set of auxiliary convolutional layers has been implemented in order to extract the feature for multiple scales and progressively decrease the
size of the input for each following layer.

The snippet in Figure 2 shows a Python pseudocode showing the proposed network.

In row 2 of the snippet in Figure 2 we load the base model i.e., the MobileNetV2 network with pre-trained ImageNet weights. ImageNet is a large database of images, created for use, in the eld of computer vision, in the eld of object recognition. The dataset consists of more than 14 million images that have been manually annotated with the indication of the objects they represent and the bounding box that delimits them. This is one of the advantages of using transfer learning, that is to "inherit" a trained network with a very large dataset of generic images to create a specialized model on a more speci c task. In our case the generic task (performed by the MobileNetV2 network) represents the generic object detection from image, while the speci c one is the detection of people wearing (or not wearing) a face mask.

Rows from the 6 to the 11 are related to the layers we added for the specialised task. In detail we consider following layers:

- **AveragePooling2D**: pooling is basically the task related to "downscaling" the image obtained from the previous layers. It can be compared to shrink-ing an image to reduce its pixel density;

- **Flatten**: reshape the tensor to have a shape that is equal to the number of elements contained in the tensor;

- **Dense**: represents a normal layer consisting of n neurons (in this case from 128), in practice the classic scheme of the arti cial neural network in which the inputs are weighed and together with the bias are transferred through the activation function to the output;
• Dropout: as a regularizer which randomly sets half of the activations to the fully connected layers to zero during training. It has improved the generalization ability and largely prevents overfitting;

• Dense: in this case we consider as final layer a Dense layer with 2 neurons (one for the mask prediction and the second one for the no mask prediction).

With the last row in the snippet shown in Figure 2 we place the layers we added on the top of the MobileNetV2 model, in this way we consider the MobileNetV2 network (with its training on the ImageNet dataset) with the additional five layers we above described, aimed to make a binary prediction (as shown from the last Dense layer with 2 neurons). Table 1 shows the details for the layers we added in terms of output shape and parameters.

| Type            | Output Shape   | Parameters |
|-----------------|----------------|------------|
| Average Pooling2D | (1, 1, 1280)  | 0          |
| Flatten         | (None, 1280)   | 0          |
| Dense           | (None, 128)    | 163968     |
| Dropout         | (None, 128)    | 0          |
| Dense           | (None, 2)      | 258        |

Once generated the model with the network architecture we designed, we store the model we obtained. This model represents the knowledge of the proposed approach for face mask detection and localisation. The storing of the generated model terminates the Training step.

Once the model is stored, in the Validation step we test the model effectiveness. The model is loaded into the memory and, subsequently, we detect whether in the images/video streams there is/are faces (i.e., Face detection in real-time images/video stream). If the model founds some face, it draws the region of interest (ROI) (Face ROI extraction) around the face. In this way the model can focus only on part of the image/video under analysis relating to a face avoiding the rest of the image/video under analysis. We highlight that the proposed model is able to detect even more faces contextually present in the same image/video stream. Once we marked only the parts of the image related to faces, these parts are the input to the model that will output a prediction for each face images (Model Prediction in Figure 1). The model outputs a certain prediction (i.e., mask or not mask) with a certain probability (from 0 to 100%). Thus, we draw on the input images/video stream the ROI, the label and the prediction probability (to understand the degree of confidence with which the model has predicted a certain label) and the results (i.e., the images/video with ROI, label and probability prediction) are stored.

We consider the proposed method explainable for its ability to automatically draw the ROI in the image symptomatic of a certain detection (i.e., by using

https://mc.manuscriptcentral.com/jamia
Figure 3: An example of detection.

the bounding box). In this way it is possible to visualise the area of the image under analysis responsible for a certain prediction, in order both to evaluate the effectiveness of the proposed model (to understand whether it is correctly considering a person in the image under analysis), but also to automatically identify the subject who is by carrying out the violation (in case the mask is not correctly worn).

3 Experimental Analysis

In this section we describe the experiment we conducted in order to demonstrate the effectiveness of the proposed method.

We obtained a dataset composed by 2165 images represented people with face mask and by 1930 images related to people without face mask for a total of 4095 images. The images were obtained from the RMFD dataset\(^1\) and from a Kaggle repository\(^2\)

From the implementation point of view, we consider the Python programming language, Tensor ow\(^3\) (the Google library for artiﬁcial intelligence experiments providing a plethora of supervised and unsupervised algorithms) and Keras (a library for neural networks management). Keras works as an interface at a higher level of abstraction than other similar lower level libraries, and sup-ports TensorFlow as a back-end. The machine used to run the experiments and to take measurements was an Intel Core i7 8th gen, equipped with 2GPU and 16Gb of RAM and Microsoft Windows 10 as operating system. For research purposes the source code developed by authors is freely available \(^4\), providing also the model generated by the proposed deep learning network.

Figures 3 and 4 shows two examples different of the detection provided by the proposed approach.

\(^1\)https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset

\(^2\)https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset

\(^3\)https://www.tensorflow.org/

\(^4\)https://mega.nz/file/AM93lKhA#nOryc32RZV1oYjAJ9hnTFp0Lv1vMEfrib3K-NDulw
As shown from the example in Figure 3, the proposed approach is able to correctly detect different type of face mask, as a matter of fact in Figure 3 both the masks (i.e., the white and the black ones) are detected with a probability equal to 100%. The aim of this example is to demonstrate that the proposed method is resilient to the type and color of the mask (in fact also the mask model is different in the face masks in Figure 3).

The second example in Figure 4 is aimed to demonstrate that the proposed is resilient to the facial expression. As a matter of fact we consider an image related to a woman with six different facial expression and the related facial expressions with the face mask: the proposed approach reaches successfully to detect all the sub images where the face mask is worn.

In Figure 5 another example of detection is shown.

Differently from the detection example shown in Figure 4, in Figure 5 we shown the detection obtained from different people wearing different face masks. It is interesting to highlight that also when people are using face mask with different colors, the proposed method is able to detect the face mask correctly.

In Figure 6 we show an example of frame obtained from a video stream, with the aim to show that the proposed method can effectively be embedded into video surveillance cams.

As show from the frame shown in Figure 6 the proposed method is able to detect all the three people wearing the face mask. In particular the proposed method is able to detect the girl on the left with her face slightly lowered. In the frame there are also several people not wearing the face mask: as evidenced from the red bounding box in Figure 6 the proposed method is able to rightly identify the infractions.

To assess the effectiveness of the proposed approach we take into account the accuracy and the loss metrics.

The accuracy is defined as closeness degree related to the measurements of a quantity to that quantity’s true value: it is basically the fraction of the
Figure 5: A third example of detection.

Figure 6: A fourth example of detection.
predictions that are correct and it is computed as the sum of true positives and negatives divided all the evaluated images:

\[
\text{Accuracy} = \frac{tp + tn}{tp + fn + fp + tn}
\]

where \(tp\) indicates the number of true positives, \(tn\) indicates the number of true negatives, \(fn\) indicates the number of false negatives, \(fp\) indicates the number of false positives.

The loss metric represents a quantitative measure of how much the predictions differ from the assigned label. For definition it is inversely proportional to the model correctness.

The loss interpretation is how well the model is doing for the training and validation sets: it is basically a summation of the errors made for each image in training or validation sets.

In a nutshell, the supervised learning of a neural network is done like any other machine learning: a training dataset is presented to the network, the net-work output is compared with the desired output, an error vector is generated, and corrections are applied to the network based on it, usually using a back propagation algorithm. The groups of training data that are treated together before applying the corrections are called epoch (epochs). We set the epoch number equal to 20. In Figure 7 there are the accuracy and the loss trends for the training and for the evaluation step.

Table 2 shows the results we obtained for each epoch: with the label train loss we indicate the loss values for the training, with the val loss the loss values for the evaluation, with train acc the accuracy values for the training step and with val acc the values of the accuracy for the evaluation step.

In the last epoch i.e., the 20-th one, with regard to the evaluation task the accuracy we obtain is equal to 0.98 and the loss equal to 0.03.

In Figure 7 we shows the trends for the accuracy and loss for the training and the evaluation steps. On the x axis there is the number of considered epochs (from 0 to 20), while on the y axis we represent the accuracy and loss values for a certain loss (ranging from 0 to 1). From an ideal point of view, we expected an accuracy equal to 1 and a loss equal to 0.

As shown from Figure 7 the accuracy trends are increasing while the loss one is decreasing. This is symptomatic that, along the 20 epochs, the network is learning the distinctive features of mask and not mask images.

We also provide details about the time performance analysis i.e., the time required by the proposed method to generate the mask/ no mask detection with the related green/ red bounding box. On the machine considered for the experimental analysis in average the proposed method employed 4.7 seconds to process a never seen image.
| Epoch | train loss | train acc | eval loss | eval acc |
|-------|------------|-----------|-----------|----------|
| 1     | 0.5491     | 0.7650    | 0.1467    | 0.9756   |
| 2     | 0.1887     | 0.9498    | 0.0839    | 0.9841   |
| 3     | 0.1072     | 0.9702    | 0.0636    | 0.9829   |
| 4     | 0.0783     | 0.9797    | 0.0606    | 0.9853   |
| 5     | 0.0620     | 0.9851    | 0.0527    | 0.9853   |
| 6     | 0.0611     | 0.9833    | 0.0513    | 0.9866   |
| 7     | 0.0670     | 0.9828    | 0.0475    | 0.9866   |
| 8     | 0.0513     | 0.9879    | 0.0435    | 0.9853   |
| 9     | 0.0529     | 0.9841    | 0.0432    | 0.9878   |
| 10    | 0.0555     | 0.9835    | 0.0427    | 0.9866   |
| 11    | 0.0537     | 0.9852    | 0.0393    | 0.9878   |
| 12    | 0.0338     | 0.9899    | 0.0409    | 0.9878   |
| 13    | 0.0467     | 0.9851    | 0.0388    | 0.9890   |
| 14    | 0.0509     | 0.9825    | 0.0356    | 0.9866   |
| 15    | 0.0315     | 0.9930    | 0.0374    | 0.9878   |
| 16    | 0.0374     | 0.9882    | 0.0377    | 0.9902   |
| 17    | 0.0312     | 0.9914    | 0.0354    | 0.9878   |
| 18    | 0.0355     | 0.9877    | 0.0382    | 0.9902   |
| 19    | 0.0314     | 0.9908    | 0.0379    | 0.9902   |
| 20    | 0.0299     | 0.9911    | 0.0351    | 0.9890   |

Table 2: Experimental analysis evaluation.
4 Discussion

In recent times, given the COVID-19 pandemic, the use of a face mask has become essential, and consequently researchers have also begun to study how to automatically detect this type of violations but, at least at the writing time, in literature there are still not very many contributions for the moment. In this section we report the e orts produced by research community in this context also with the aim to highlight the novelty of the proposed contribution.

For instance, authors in [14]. They exploit articial intelligence, in particular the ResNet50 deep model in combination with the Support Vector Machine machine learning classier to predict if in an image under analysis people are wearing or not a face mask. The di erence between this work and our proposal is represented by the model adopted: as a matter of fact we consider a transfer learning approach based on the MobileNetV2 model for fake mask detection from images and video streams, making our method able to run also on device with limited resources for instance, mobile device but also webcam. Thus, the proposed method is applicable to each type of devices.

Researchers in [15] experiment the InceptionV3 deep learning model for face mask detection. The e ectiveness of this method is evaluated using a dataset composed by 1570 images , 785 related to people wearing a face mask and 785 images of people without the face mask, while we evaluate the proposed with a total 4095 di erent images. Moreover, the proposed approach is able to
generated the prediction also on mobile and embedded platforms.

Chen et al. [16] propose the adoption of machine learning techniques, in particular the K-nearest neighbors supervised classification algorithm for the discrimination between people wearing and not wearing the face mask. They reach an accuracy equal to 0.87, while the proposed method obtains an accuracy equal to 0.98.

5 Conclusion and Future Work

The COVID-19 pandemic has radically changed our habits. While waiting for the spread of the virus to decrease, it is necessary to observe a series of rules, such as the use of face masks in public places. The aim of the following work is to provide a method for the automatic detection of violations of persons not wearing the face mask. In detail our approach relies on the adoption of transfer learning to detect in images and video streams whether there is the presence of people wearing and not wearing the face mask. The proposed method is able to work also on devices with limited computation capabilities as, for instance, smart-phones or webcams, making the proposed approach actually implementable in a real-world context.

With regard to the future research lines, we plan to enforce the proposed approach by exploiting a series of transfer learners with the aim to try to increase the performances. Moreover, we plan to adopt the activation maps to highlight the areas symptomatic of the detection, making the proposed approach more interpretable [17, 18]. As a matter of fact, while the proposed method is able draw the bounding box around the people face: activation maps can be helpful to detect which part of the face contributes to the detection. In this way it will be possible a ner-grain detection.

FUNDING STATEMENT
This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors

CONFLICT OF INTEREST STATEMENT
The authors have no competing interests to declare

CONTRIBUTORSHIP STATEMENT
Francesco Mercaldo and Antonella Santone contributed to the design of the proposed method, to the experimental analysis and to all the aspects of the paper (from draft writing to the paper final approval).

DATA AVAILABILITY STATEMENT
The Python source code underlying this article are available in at the following url: https://mega.nz/file/AM93lKhA#nOryc32RZV1oYjAj9hnTPoLv1vMEfrigbI3K-NDulw. The datasets were derived from sources in the public domain: RMFD, https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset and Kaggle, https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset.
References

[1] Biying Hu, Shaoying Huang, and Lianghong Yin. The cytokine storm and covid-19. Journal of medical virology, 93(1):250{256, 2021.

[2] Faizan Ahmed, Syed Ahmad Chan Bukhari, and Fazel Keshtkar. A deep learning approach for covid-19 8 viral pneumonia screening with x-ray im-ages. Digital Government: Research and Practice, 2(2):1{12, 2021.

[3] Luca Brunese, Fabio Martinelli, Francesco Mercaldo, and Antonella Santone. Machine learning for coronavirus covid-19 detection from chest x-rays. Procedia Computer Science, 176:2212 { 2221, 2020.

[4] Tuan D Pham. Classi cation of covid-19 chest x-rays with deep learning: new models or ne tuning? Health Information Science and Systems, 9(1):1{11, 2021.
[5] Laura Mieth, Maike M Mayer, Adrian Hohmann, Axel Buchner, and Raoul Bell. Do they really wash their hands? prevalence estimates for personal hygiene behaviour during the covid-19 pandemic based on indirect questions. BMC public health, 21(1):1{8, 2021.

[6] M Kavitha, T Jayasankar, P Maheswara Venkatesh, G Mani, C Bharatiraja, Bheki Sipho Twala, et al. Covid-19 disease diagnosis using smart deep learning techniques. Journal of Applied Science and Engineering, 24(3):271{277, 2021.

[7] SM Didar-Ul Islam, Prantor Kumar Mondal, Nathanael Ojong, Md BodrudDoza, Md Abu Bakar Siddique, Moazzem Hossain, and Mohammed A Mamun. Water, sanitation, hygiene and waste disposal practices as covid-19 response strategy: insights from Bangladesh. Environment, Development and Sustainability, pages 1{22.

[8] H Swapna Rekha, Himansu Sekhar Behera, Janmenjoy Nayak, and Bighnaraj Naik. Deep learning for covid-19 prognosis: A systematic review. Intelligent Computing in Control and Communication, pages 667{687, 2021.

[9] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4510{4520, 2018.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248{255. Ieee, 2009.

[11] Preeti Nagrath, Rachna Jain, Agam Madan, Rohan Arora, Piyush Kataria, and Jude Hemanth. Ssdmnv2: A real time dnn-based face mask detection system using single shot multibox detector and mobilenetv2. Sustainable Cities and Society, page 102692, 2020.

[12] Carlos Ulloa, Dora M Ballesteros, and Diego Renza. Video forensics: Identifying colorized images using deep learning. Applied Sciences, 11(2):476, 2021.

[13] Rezowan Hossain Ferdous, Md Murshedul Arifeen, Tipu Sultan Eiko, and Shamim Al Mamun. Performance analysis of different loss function in face detection architectures. In Proceedings of International Conference on Trends in Computational and Cognitive Engineering, pages 659{669. Springer, 2021.

[14] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N Taha, and Nour Eldeen M Khalifa. 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, 2020.
[15] G Jignesh Chowdary, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. Face mask detection using transfer learning of inceptionv3. In International Conference on Big Data Analytics, pages 81–90. Springer, 2020.

[16] Yuzhen Chen, Menghan Hu, Chunjun Hua, Guangtao Zhai, Jian Zhang, Qingli Li, and Simon X Yang. Face mask assistant: Detection of face mask service stage based on mobile phone. arXiv preprint arXiv:2010.06421, 2020.

[17] Luca Brunese, Francesco Mercaldo, Alfonso Reginelli, and Antonella San-tone. Explainable deep learning for pulmonary disease and coronavirus covid-19 detection from x-rays. Computer Methods and Programs in Biomedicine, page 105608, 2020.

[18] Giacomo Iadarola, Fabio Martinelli, Francesco Mercaldo, and Antonella Santone. Towards an interpretable deep learning model for mobile malware detection and family identification. Computers & Security, page 102198, 2021.
Figure 1: The workflow of proposed approach.
```python
#loading the MobileNetV2 model
baseModel = MobileNetV2(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))
#building the head of the model and place it on top of the MobileNetV2 model
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
# place the model on top of the MobileNetV2 model
model = Model(inputs=baseModel.input, outputs=headModel)
```

Figure 2: Python pseudocode of the proposed network.
Figure 3: An example of detection.
Figure 4: A second example of detection.
Figure 5: A third example of detection.
Figure 6: A fourth example of detection.
Figure 7: Experimental analysis results.