UDC 004.852

**V. Oliinyk, A. Ryzhiy**

# AN EFFICIENT FACE MASK DETECTION MODEL FOR REAL-TIME APPLICATIONS

*Abstract*: Recent COVID-19 pandemic demonstrates increasing importance of facemasks and other protective equipment raising demand for specialized monitoring and control systems.

In this paper, we propose an efficient and reliable method based on Dual Shot Face Detector in order to address the problem of the masked and non-masked face recognition process. The main goal of the research is to locate people without a facemask and this task combines both finding faces and facemask recognition. In addition, multiple faces can be tracked in real-time on a video stream. Experimental results show high detection performance with Mean Average Precision of 90%. Proposed solution is stable to the change of face positions, rotations and inclines. It also allows controlling correct placement of a mask on a face.

In addition, different feature encoders were studied to find the balance between accuracy and inference time that is important for real-time performance with different hardware.

*Keywords:* facemask detection, Dual Shot Face Detector, object detection, real-time systems, computer vision.

## Introduction

The usage of personal protective equipment (PPE) is an integral part of many areas of human life. In particular, facemasks are widely used to protect people from air pollution and contamination in construction, industry and agriculture and to protect against viruses and pathological bacteria. In addition, in the context of the COVID-19 pandemic, virtually every adult on the planet faced the need to wear facemasks. According to the recommendations of the World Health Organization, wearing masks in public places is necessary to stop the spread of infection.

Obviously, there is a large number of applications that require systems for monitoring and control of PPE usage. The task of identifying people without a facemask based on a video data or images can be a typical example of such application. The widespread use of security cameras in public places provides enough data for monitoring and control without any extra cost. However, the problem can still be complicated: a large number of people in the image, their different positions, rotations and inclinations, scale, occlusion and so on. Therefore, creation of highly efficient and affordable real-time video processing systems is a challenging and actual goal in the PPE detection and control tasks.

The problem of object detection in video can be presented as a task of image processing considering single video frames as a separate source of data. Definitely, processing



 ISSN 1560-8956

sequences of frames and tracking objects using visual trackers [1] can further improve quality and reliability of the system. However, tracking systems are more complicated, resource consuming and specialized systems, so tracking methods will not be the subject of this article. We will focus on the problem of finding people without masks in the images. Demonstration software will still be tracking faces using OpenCV (other computer vision framework may be used instead) but information from sequences of frames will not be used to improve detection accuracy.

The task of object detection is to find and mark a rectangle over the object of a certain class in a given picture. Object detectors based on variations of convolutional neural networks are typically used for finding objects in an image/video in modern integrated AI systems. Such models are usually used with multiclass detection problems while finding facemasks on the people can be formulated as a single class task. Moreover, the task of finding people without masks is very similar to the faces detection problem. Therefore, it is reasonable to build facemask detection model based on a specialized state-of-the-art face detector model.

## Related research and publications

The most straightforward way of solving facial mask recognition task is based solely on image classification [2, 3]. Such approaches require finding and cropping faces from original images that can be done with a separate detector. This combination of detector and classifier usually works worse and longer than a fully trained detector.

Another approach consider usage of R-CNN family [4] (R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN), YOLO family [5] (YOLO, YOLO v3, YOLO v4, YOLO X), SSD [6] and other similar architectures for fast but low quality detection. This approach has also been popular in face detection for quit a long time due to the speed of interference and good result when applied to real-time video processing. For example, fast in browser ‘WearMask detector’ model was proposed in the paper [7] that uses the YOLO architecture to find people with and without masks. The authors offer a method of finding masks in real time using high-performance neural network inference computing framework and a stack-based virtual machine that shows good Average Precision when IoU is 0.5 (mAP@0.5) of 0.89.

Since the task of finding faces without PPE is very similar to the well-known problem of just finding faces we propose to use face detector model. We will use the state-of-the-art Dual Shot Face Detector [8] to build our model and conduct a series of experiments with it.

This deep learning model contains additional Feature Enhance Module and uses Progressive Anchor Loss, which is calculated by two different sets of anchors, to effectively facilitate functions. Finally, Improved Anchor Matching (IAM) is used to provide better initialization by integrating a new anchor assignment strategy into the augmentation data.

**Overview of proposed Mask Detection System**

We will use images to detect people without PPE. These images will be obtained from the video either real-time or recorded by splitting it into frames. We will not use any kind of additional information about sequences of frames like tracking algorithms [1] etc. Feeding selected images to the neural network will result in bounding boxes of people faces without masks. There can be several persons in one frame and the detector will find all the people without masks in the frame. Proposed detection system can optionally also find people wearing a mask.

General workflow of the system is shown in Fig. 1. First, the video is divided into frames. Then the neural network processes each frame sequentially and provides probabilities and bounding boxes coordinates for each class. Finally, if the probability of the unmasked person class exceeds 0.5, an alert signal is triggered.

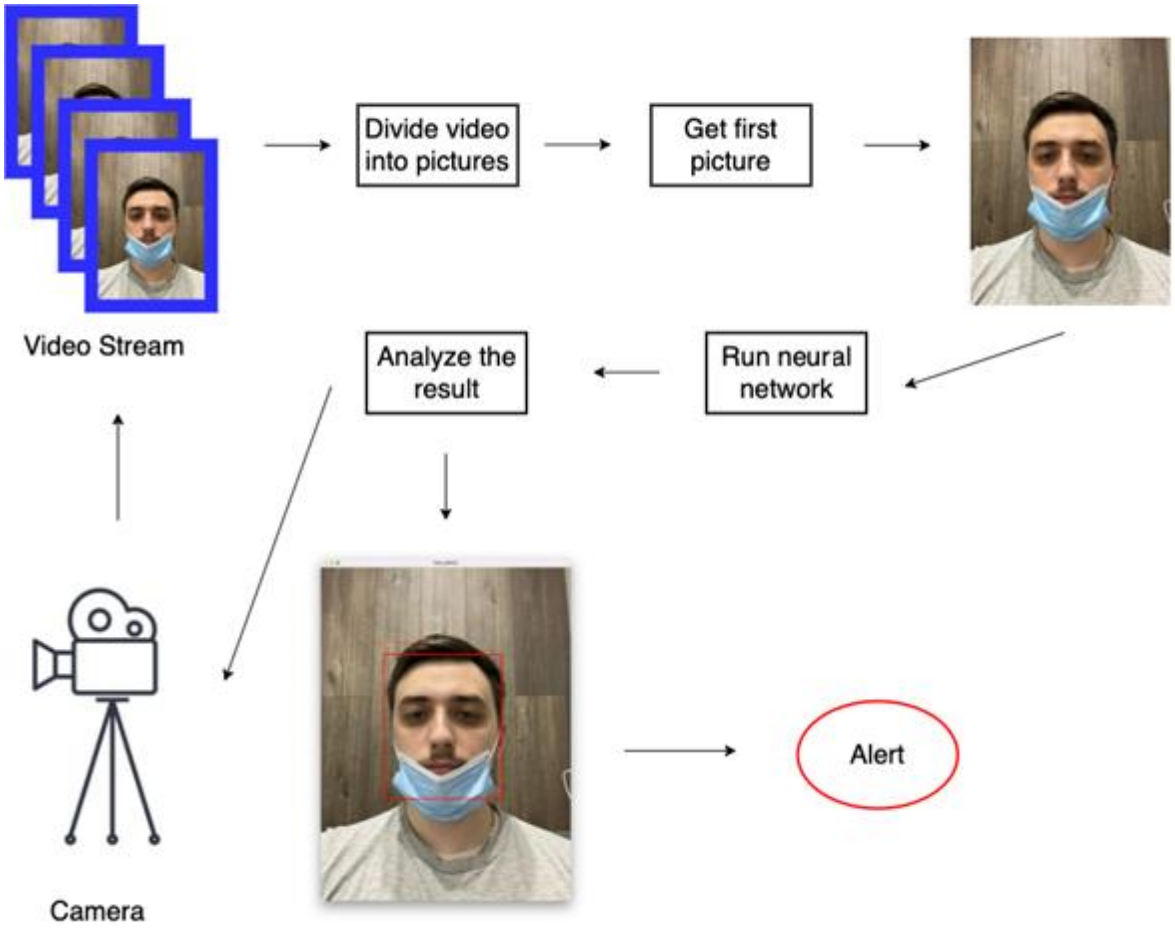


*Figure 1.* Workflow diagram of the proposed Mask Detection System


ISSN 1560-8956


*Figure 2.* Sample images from RMFD dataset (left) and SMFD dataset(right)

Considering the fact that neural network performance may be lower than frame rate of the video stream, we will not process every frame. Detection system will consider only frames that are available immediately after it has finished its current processing.

We will additionally study the impact of backbone network on processing time and consider its application in real-time solutions.

## Dataset preparation

Quality of the first facial mask detectors was naturally restricted by the lack of specialized datasets at the beginning of the COVID-19 pandemic. Therefore, many researchers artificially added masks to available datasets with people's faces. As a result, many existing datasets for face detection were adopted to the problem of masked face detection. SMFD dataset [9] is a popular artificial dataset created that way. Today we also have collections of real photos available. However, our experiments show that increasing the amount of training data with such augmented datasets provides increase in accuracy, so we will also use SMFD dataset.

In order to compare received results with actual state-of-the-art models we will use the dataset and metrics from [7]. The final dataset for mask detection will be a compilation of available datasets and will contain two categories of datasets: real faces with real masks (MAFA [10], RMFD [11], MMD) and real faces with generated masks (SMFD). In general, 9 097 images with 17 532 labeled boxes were divided into 80% training and validation and 20% testing datasets.

Sample images from the datasets that were used to train our model are shown in Figure 2.

## Proposed model and implementation

Object detection network will be used to solve the problem of finding faces without masks. We will use latest architecture DSFD: Dual Shot Face Detector [8] as a base model. This network, that shows state-of-the-art results in face detection, is similar in structure to the

Single Shoot Detector [6]. Unlike SSD, it has two feature layers instead of one: the original SSD layer and additional one for improving features with the Feature Enhance Module. This Module is able to enhance original features to make them more discriminable and robust by normalizing feature maps with 1x1 convolutions and getting element-wise product with up-sampled upper feature map. Finally, feature maps are split to three sub-networks containing different numbers of dilation convolutional layers.

The backbone network (feature encoder) is a variable part of typical detection network including DSFD. Larger network candidate will result in higher accuracy, but a smaller one works faster. Proper choice of this encoder network allows balancing between detection quality and speed. Real-time applications provide restrictions to performance time therefore selection of the backbone network is studied in this paper and results are shown and discussed in the next section.

Proposed network was trained using loss function consisting of two components:

• $L_{loc}$ is the part of the loss function responsible for localizing the bounding box. This is the L1 loss between the expected and the actual box positions. These parameters include offsets for the center point, width and height of the bounding box.

• $L_{class}$ is the second part of the loss function responsible for classifying a particular bounding box. This is a typical classifier with a softmax function for a multiclass or a sigmoid for a single class task.

Mean Average Precision with IoU threshold=0.5 was chosen as the main metric.

The neural model was designed, implemented and studied in Tensorflow 1 framework using Python. Numpy, Pandas and OpenCV libraries were used for research and data pre-processing. Neural model was trained using Momentum Optimizer algorithm with *lr* = 0.0001 and *batch size* = 16 on NVidia RTX 2080ti.

The height and width of all images was constant and equal to 320 pixels.

## Experimental research and results

### *Experiments*

We started with a basic MobileNetV2 with *width multiplier* = 1.4 as a feature extraction network. It was trained on our prepared combination of datasets (see Dataset preparation section) for a single class of non-masked persons and resulted mAP = 0.6767 on validation dataset for this setting.

The result is very low, so we took other version of MobileNetV2 which was pre-trained on ImageNet. Despite the fact that transfer learning is a common and widely used technique for image processing, the effect of its application depends on the type of objects that are classified.

 ISSN 1560-8956

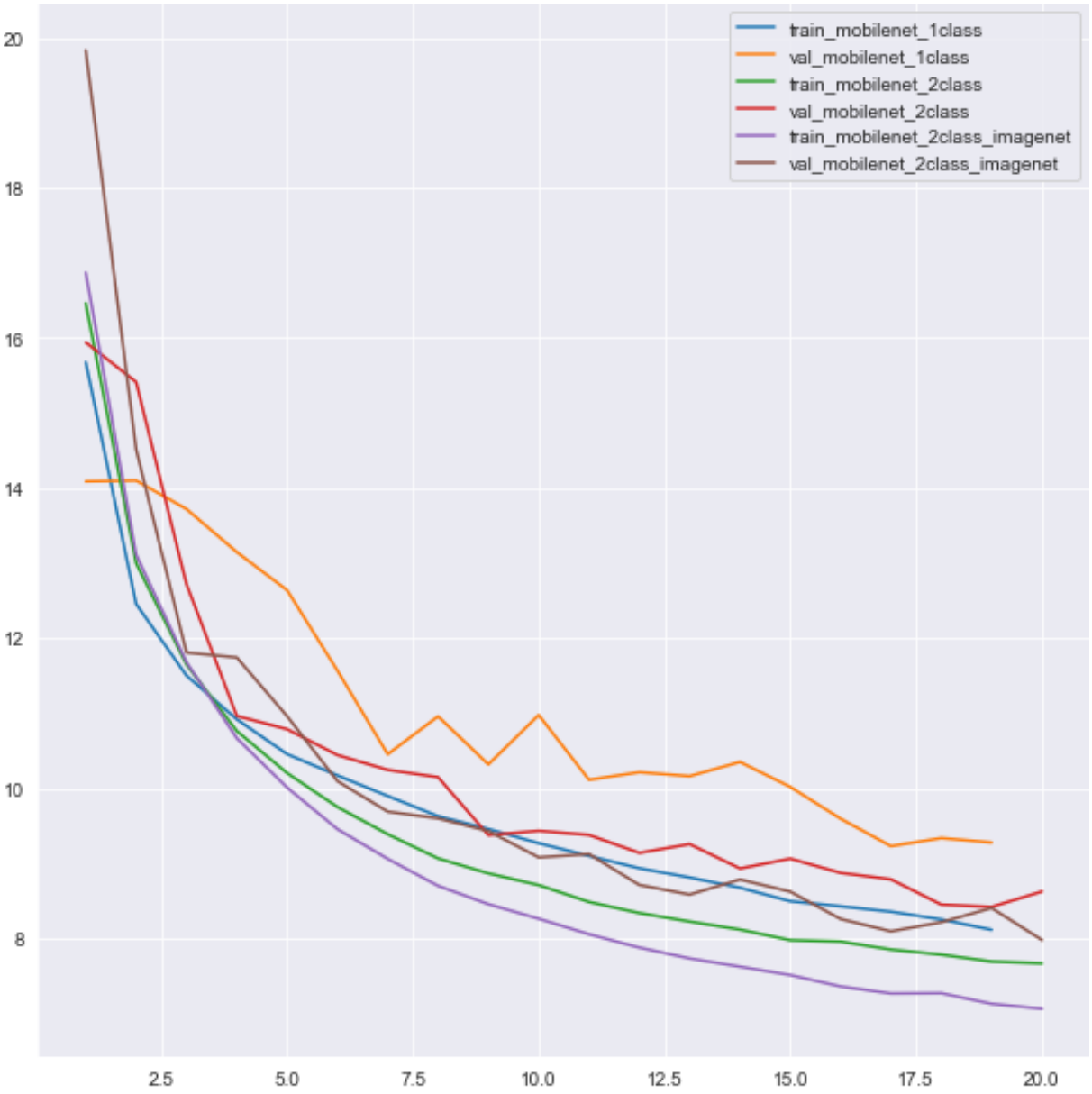


*Figure 3.* Values of total loss on train and validation datasets for each network setup

As expected, the neural model with pre-trained parameters showed higher results after training. Its mAP value on the validation dataset became equal to 0.7747.

After that, we took all available data and solve two-class classification problem, detecting people wearing masks and without masks. Performance metrics were calculated separately for each class, and only values of the class without masks were taken to consideration to make comparison with previous experiments.

Following result on the validation dataset was received: mAP = 0.7899 for the ‘without a mask’ class and mAP = 0.9020 for the ‘with a mask’ class.

Values of total loss for each experiment are shown in the graph in Figure 3.

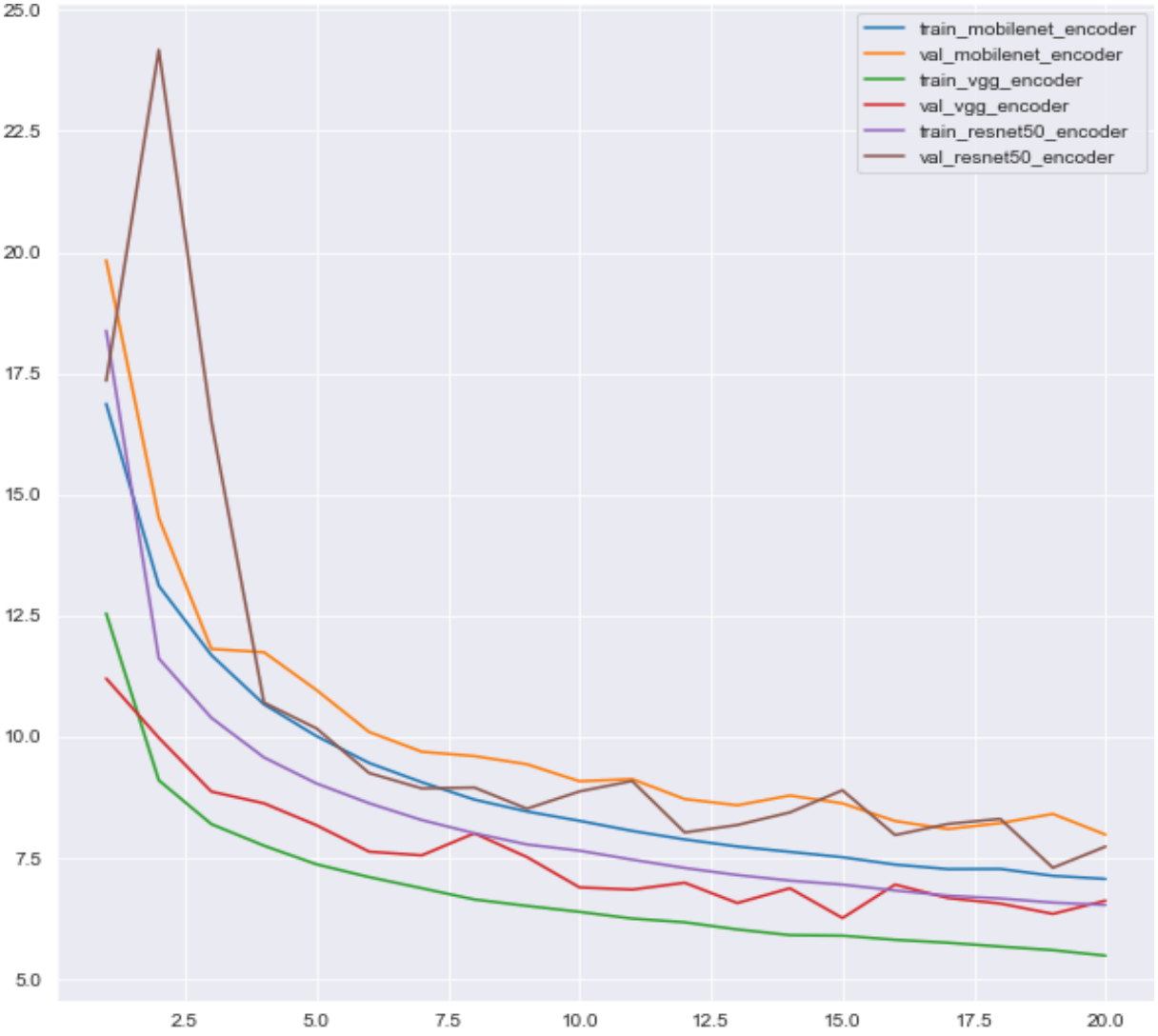


*Figure 4.* Comparison of total loss on train
and validation datasets for different feature extraction networks

Since the speed and efficiency of the network highly depends on the feature extractor network, we decided to compare most popular networks that are usually used in detector architectures.

Previous results were taken to consideration so all further experiments were based on a two-class training setup with pre-trained networks. We made experiments with: MobileNetV2, ResNet50 and VGG16. Total loss comparison charts can be found in Figure 4.

The experiment with VGG showed the best result (lowest value of total loss) and the MobileNetV2 model was the worst in terms of accuracy. However, working with real-time video requires adequate performance time. Therefore, it would be wise to compare the inference time of each model.

 ISSN 1560-8956

*Table 1.*

**Inference time of mask detection network for different feature extractors**

| Feature extractor | Average inference time (in seconds per inference) |
|---|---|
| MobileNet | 0.1 |
| Resnet50 | 0.16 |
| VGG16 | 0.23 |

We used TensorFlow V1 without any optimization to measure performance time. The results (see Table 1) confirms the assumption that more complex models with higher accuracy has lower inference time.

The MobileNetV2 encoder is the best in terms of performance, while VGG16 is more than 2 times slower. Inference time obviously depends on productivity of the available GPU resources and this difference may not greatly affect overall performance of the mask detection system. However, real-time applications and practical aspects of making cheap solutions often raise restrictions on maximum processing time. Therefore, the more powerful resources we have, the larger model may be used and the better accuracy can be achieved.

The results as well as a comparison with previous results are shown in the Table 2.

*Table 2.*

**Mean Average Precision (mAP@0.5) for all models and cases**

| Experiment | Validation mAP for the class without a mask | Validation mAP for the class with a mask | Average validation mAP for two classes | Average training mAP for two classes |
|---|---|---|---|---|
| *Base model* | 0.6767 | - | 0.6767* | 0.7108* |
| *Pretrained (ImageNet) base model* | 0.7747 | - | 0.7747* | 0.8006* |
| *DSFD (MobileNetV2 encoder)* | 0.7899 | 0.9020 | 0.8459 | 0.8630 |
| *DSFD (Resnet50 encoder)* | 0.8126 | 0.9024 | 0.8575 | 0.8777 |
| *DSFD (VGG16 encoder)* | 0.8669 | 0.9361 | 0.9 | 0.9282 |

* Base model and its pre-trained version is a DSFD with MobileNetV2 encoder trained on single class so we consider averages to be same as the main results

***Comparison with previous works***

The authors of previous works used different performance metrics depending on their tasks and trained their models on different datasets. However, some of them also measure the

accuracy of the resulting bounding boxes, in particular Mean Average Precision with IoU threshold set.

We run experiments on a dataset from [7] using the original split into a training and validation sets to make a proper comparison. After that, we measured mAP@0.5 for the two-class problem: people without a mask and people with a mask, by taking average result for both classes. According to the results, the best mAP@0.5 of our model on the validation dataset was 0.9, which is 1% better than in [7].

Such results shows that DSFD based model outperforms tuned version of general purpose network models like R-CNNs, YOLO and SSD and proves the concept of using face detectors as a basis for such tasks.

***Examples of use***

We have developed demonstration program based on OpenCV framework that captures video stream from a camera and uses proposed detector to process it. Examples of the results are shown in figures 5 and 6. Several tests was made to check robustness of the model depending on the position of the head (Fig. 5) and incorrect usage of the facial mask (Fig. 6)

*Figure 5.* Testing proposed model with different face positions

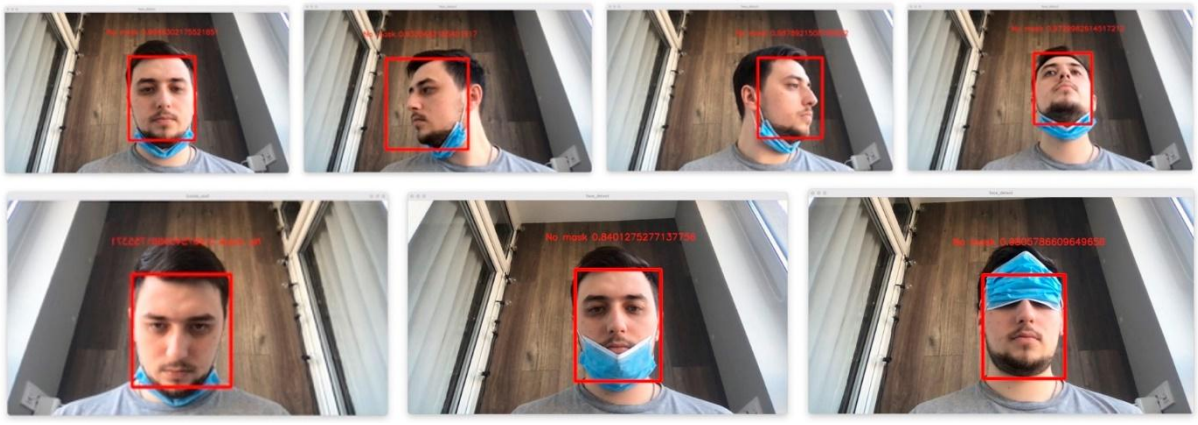

*Figure 6.* Testing proposed model with incorrect mask usage


ISSN 1560-8956

## CONCLUSION

In this paper, we proposed a model based on a DSFD architecture to solve the problem of facial mask monitoring and control. We proved the concept that facial detectors outperforms general detectors for this task. The model was implemented in a demo control system capable to detect multiple people without mask or those using it in inadmissible way. Experiments confirmed high accuracy compared to the modern state-of-the-art models, sustainability to the changes of face position. The model is well adopted to real-time processing even with the largest tested VGG16 feature encoder. In addition, the proposed model can easily be adopted to various resource-limited applications such as video surveillance, monitoring or control as a separate service or embedded solution by using smaller encoder as shown above.

Future work will be focused on tuning performance of the model and implementing it to our AR-solution [12, 13].

## REFERENCES

1. Олейник В.В. Метод визуального мультитрекинга в реальном времени на основе корреляционных фильтров / А.С. Пантелеев, В.В. Олейник // Міжвідомчий науково-технічний збірник "Адаптивні системи Автоматичного Управління", К: Політехніка. – 2018. – Т.1, №32. – Сс. 97-106.

2. Walid Hariri. Efficient Masked Face Recognition Method during the COVID-19 Pandemic / arXiv:2105.03026. – 2021.

3. Mandal B. Masked Face Recognition using ResNet-50 / Bishwas Mandal, Adaeze Okeukwu, Yihong Theis // arXiv preprint ArXiv:2104.08997. – 2021.

4. He K. Mask R-CNN / He K., Gkioxari G., Dollár P., Girshick R. // In: IEEE international conference on computer vision. – 2017. – Pp. 2961–2969.

5. Zheng Ge. YOLOX: Exceeding YOLO / Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, Jian Sun // arXiv preprint arXiv:2107.08430. – 2021.

6. Liu W. SSD: Single Shot MultiBox Detector / Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg // Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science. – 2016. – Vol 9905. Springer, Cham. – Pp. 21–37.

7. Wang, Z. WearMask: Fast In-browser Face Mask Detection with Serverless Edge Computing for COVID-19 / Wang, Z., Wang, P., Louis, P.C., Wheless, L., & Huo, Y. // arXiv preprint ArXiv:2101.00784. – 2020.

8. Li Jian. DSFD: Dual Shot Face Detector / Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, Feiyue Huang // arXiv:1810.10220v3. – 2019.

9. Dataset, https://github.com/prajnasb/observations, online accessed May 23, 2022.

10. Ge Shiming. Detecting Masked Faces in the Wild with LLE-CNNs / Shiming Ge, Jia Li, Qiting Ye, Zhao Luo // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2017. – Pp. 2682-2690.

11. Wang Z. Masked face recognition dataset and application / Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei et al. // arXiv preprint arXiv:2003.09093. – 2020.

12. Oliinyk V. Method for improving accuracy of mobile AR navigators. ISJ Industry 4.0. – 2020. – Vol. 5, Is. 1. – Pp. 21-22.

13. Олійник В.В. Алгоритм уточненого позиціонування в навігаційних системах доповненої реальності / В.В. Олійник, Є.А. Яременко// Міжвідомчий науково-технічний збірник «Адаптивні системи автономного управління», К: Політехніка, 2018. – Т.2, №33. – Сс. 56-65.

 ISSN 1560-8956