Design Flow for Real-Time Face Mask Detection Using PYNQ System-on-Chip Platform

Tianyang Fang, Xin Huang and Jafar Saniie

Embedded Computing and Signal Processing Research Laboratory (http://ecasp.ece.iit.edu)

Department of Electrical and Computer Engineering

Illinois Institute of Technology, Chicago IL, U.S.A.

***Abstract*— Study shows that mask-wearing is a critical factor in stopping the COVID-19 transmission. By the time of this article, most US states have mandated face masking in public space. Therefore, real-time face mask detection becomes an essential application to prevent the spread of the pandemic. This study will present a face mask detection system that can detect and monitor mask-wearing from camera feeds and alert when there is a violation. The face mask detection algorithm uses Haar cascade classifier (HCC) to find facial features from the camera feed and then utilizes it to detect the mask-wearing status. The detection system runs on a PYNQ-Z2 all-programmable SoC platform, where it will pipeline the camera feed through the FPGA unit and carry out the face mask detection algorithm in the ARM core. Potential delays are analyzed, and efforts are made to reduce them to achieve real-time detection. The experiment result shows that the presented system achieves a real-time 45fps 720p Video output, with a face mask detection response of 0.13s.**

I. Introduction

According to recently published research on COVID-19, every 10% increase in mask-wearing means a three-fold likelihood of stopping the breakout in a community, and other policies, such as social distancing, are not near as effective [1]. Mandating the face mask thus is urgent and essential. In fact, at the time of this article, most states in the US have mandated mask-wearing in public space by law. However, enforcing the mandate by pure human force can be tedious since the task needs continuous focus for hours and dangerous as it increases the exposure risk for the enforcer.

A face mask detector is an algorithm that detects the mask-wearing status of a person. Using a face mask detector can relieve human factors from the mask-wearing enforcing task. Another application of the face mask detector is for quite the opposite purpose. In some sensitive security zones where masking is prohibited, face mask detection can alert a masked person in the security footage. The wide range of applications of face mask detection makes it a popular research topic. A popular algorithm for face mask detection is the Haar cascade classifier (HCC) [2]. In a recent work [3], they proposed human face detection algorithm by primitive Haar cascade algorithm combined with three additional weak classifiers. The test results are efficient and achieve state-of-the-art performance by detecting people under different occlusions and illuminations and some orientations and rotations. A precise face and eye detection method using HCCs was proposed in [4]. The proposed system evaluated 10,000 test images to detect and precisely localize 94% of the eyes properly. RetinaFaceMask method was proposed [5] for a high-accuracy and efficient face mask detection. The method is a one-stage detector, which consists of multiple feature maps and a feature pyramid network (FPN) to fuse the high-level semantic information, a novel context attention module to detect face masks. Reference [6] presents a novel deep learning model for face mask detection. The research proposed a feature extraction process based on the ResNet-50 deep transfer learning model and the detection of medical face masks based on YOLO v2. In similar research [7], YOLO v3 is used and achieved real-time face mask detection. In reference [8] a model named SSDMNV2 for face mask detection using OpenCV Deep Neural Network (DNN) was proposed. The algorithm can detect faces in most orientations and provides lightweight and accurate predictions for classification based on mask-wearing. Similar research on face mask detection sometimes focuses on detecting masked faces from all faces, usually for security or even anti-terrorist aims [9]. Although the false negative in this and medical face mask detection scenarios can have different significance, the design logic shows a certain degree of resemblance.

This article proposes a face mask detection system that can detect mask-wearing status on a camera feed in real-time. When a person appears in the video without wearing a face mask, they will be marked as so. The face mask detection algorithm first abstracts facial features using the HCC then analyzes the facial features to detect the mask-wearing status. We implemented the system on a PYNQ-Z2 System-on-Chip platform, where the FPGA unit pipelines the Video stream, and the face mask detection runs on the ARM processor. Section 2 describes the face mask detection system. Section 3 presents and evaluates the system's results and performance in a real-world test scenario and an open-source database. Section 4 outlines the Conclusion and future works.

II. System Design

As shown in Figure 1, the system is implemented on a PYNQ-Z2 SoC platform. PYNQ is the abbreviation for Python productivity for Zynq. The PYNQ-Z2 board is based on the Zynq-7020 SoC device, which is integrated with a dual-core ARM Cortex-A9 processor and an Artix-7 Programmable logic. PYNQ aims to make the design process to be more efficient and at a higher level of abstraction. Instead of building the FPGA design from VHDL codes and IP cores, PYNQ boards use overlays, collections of programmable logic circuits that can be used as hardware libraries and can be called from the ARM core through APIs. Thus, we choose the PYNQ platform for the feasibility of finding overlays that suit our design and its native support for the python language, which can be convenient when programming the face mask detection algorithm using OpenCV. Using an SoC platform in general also allows great improvement potential as it is meant for prototyping and making quick changes to the design.

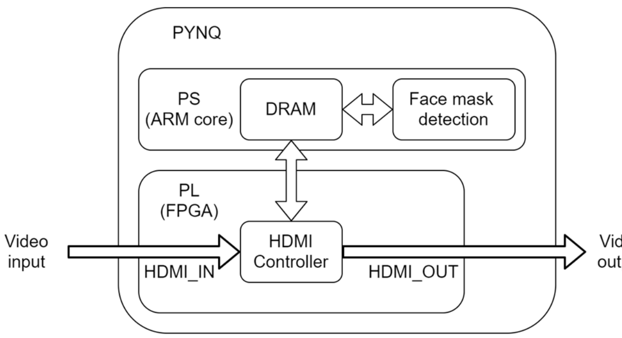

Figure 1. The proposed face mask detection system

A. Video Pipeline

As shown in Figure 1, the Pynq-Z2 default overlay provides HDMI input and output through the FPGA unit with direct memory access. This overlay is suitable for our design as the HDMI input can provide the camera feed, and the output can go to the HDMI output via the DRAM. The video pipeline can bypass the CPU and go directly through the overlay. As the CPU is not involved in the process, this method yields no software streaming delay. However, this method is not viable as the face mask detection needs to run in the ARM core.

Instead, our algorithm uses a pipeline structure that involves the ARM core to carry out the face mask detection. The delay introduced by using the ARM core to bridge the video I/O through the DRAM is measured to be $3.8ms$. For a 60fps camera feed, which has a frame delay of $16.7ms$, the ARM core delay is small enough to avoid any frame drop.

B. Face Mask Detection

For face mask detection, we choose the Haar cascade classifier (HCC) for face and facial features detection due to its efficiency and availability. HCC is a machine learning approach where the classifier is trained from positive and negative images. The training process consists of two major stages: the Haar features extraction and assignment of the Haar features into a cascade of classifiers [2].

In the feature extraction stage, three groups of features are used as shown in Figure 2. Each feature is calculated using the equation:

$$\text{feature score} = \frac{\sum W}{\sum B} \tag{1}$$

where $\sum W$ is the sum of pixels in the white rectangles, and $\sum B$ is the sum of pixels in the black rectangles. Figure 3 demonstrates two features calculated on a training image. The training image is of fixed window size, and every feature is calculated on the image to get a feature score.

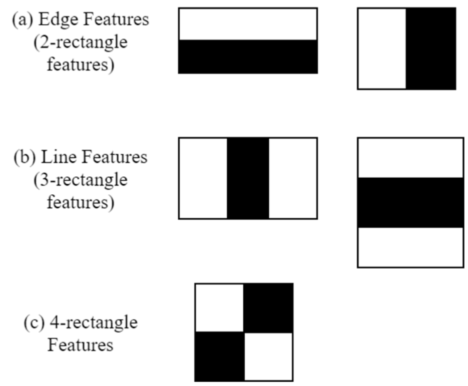

Figure 2. Haar features

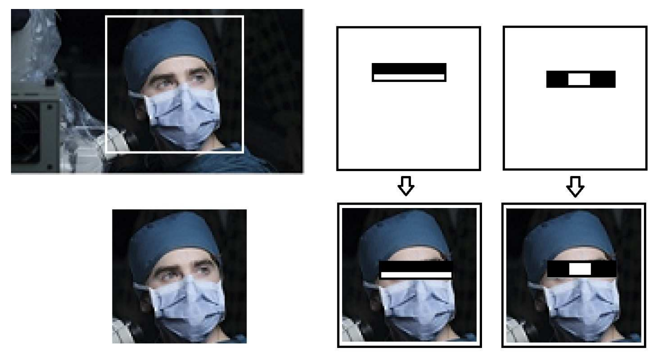
Figure 3. The feature calculation process

With all the extracted features, the AdaBoost algorithm is then applied to select a set of features that suit the classifier the best. AdaBoost stages the selection by first picking several weak classifiers that use fewer features but have less accuracy [10]. A weak classifier alone cannot classify the subject effectively, but combining a selective of weak classifiers will form a robust classifier, and this is why HCC usually can achieve above 90% detection rate.

However, the features selected by AdaBoost are still a large selection. For example, in [2], the selection consists of near 6000 features. Applying them at once for every window in a test image is time-consuming and inefficient because usually, in an image, only a few regions contain the target objects, so applying the whole classifier to an irrelevant window should be avoided. The Cascade of Classifier solves this problem by grouping the features into stages of classifiers. A window of the test image will be passed down to each stage, one at a time, and whenever a stage does not pass, this window will be discarded immediately. Only a window that passes all the stages of the classifier is marked positive. By arranging the classifiers' stages properly, irrelevant windows can be ruled out in the first several stages so that the classifier can focus on windows that contain the target object.

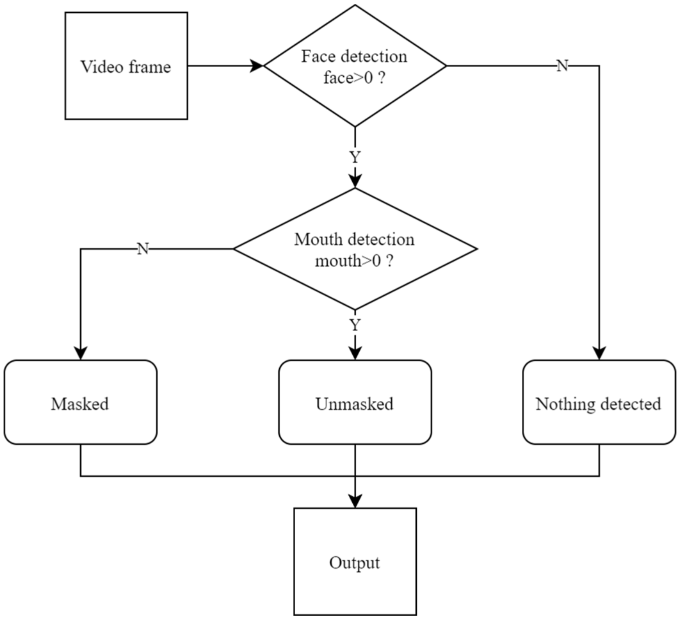

Figure 4. The face mask detection algorithm

Figure 4 shows our proposed algorithm for face mask detection. The algorithm first uses a HCC to detect faces in the input image. A Haar cascade for frontal face detection is used in this process. If no face is detected, the image will be fed straight to the output to keep the video stream. If a face is detected in the image, the face rectangle is then fed into another HCC for mouth detection, determining the mask-wearing status.

Since the HCC has a fixed window size, we need to scale the classifier to detect the faces in various scales that would appear in the camera footage. The scaling factor can be selected to achieve the desired efficiency and accuracy. However, the scaling detection window will cause false positives from the image. Figure 5 shows this issue where a pattern in the bush is recognized as a face. This problem can be countered by only marking a positive window that has a certain number of neighbors that are also positive. The selection of the scaling factor and the minimal neighbors is discussed in the evaluation.

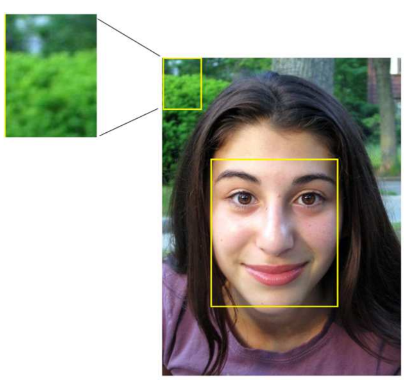

Figure 5. A false positive from multiscale detection

C. Real-Time Concerns

For practical use, achieving real-time face mask detection is essential. A large delay means the person in the footage could have left before the detector sends an alert. Every possible delay in the video pipeline, either from the face mask detection or memory maneuvers, should be considered and reduced using SoC design techniques.

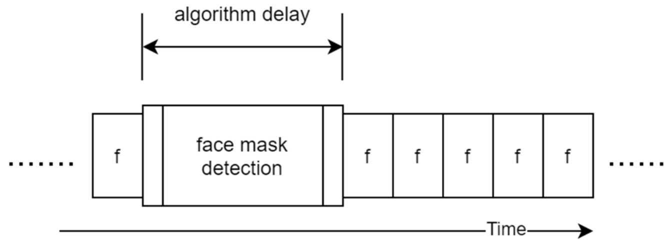

Figure 6. Running the algorithm in the main thread

One significant delay comes from running the HCC. As mentioned in the previous section, the algorithm has an average run time of $t_{detection} = 83.1ms$. Considering applying the detection continuously in a 60fps Video input, as shown in Figure 6, this leads to the frame rate dropping to

$$f = 60\left(1 - \frac{t_{detection}}{t_{detection} + t_{frame}}\right) \approx 10\text{fps} \tag{2}$$

here t_{frame} is the frame delay for a 60fps Video, thus $t_{frame} = 1/60 \approx 16.7\text{ms}$. With a 50fps frame drop, the final video output will be in 10fps.

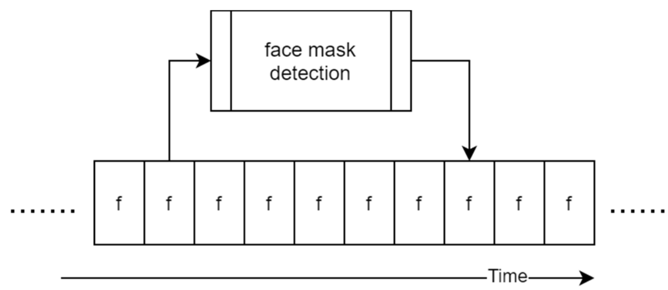

Figure 7. Running the algorithm in a second thread

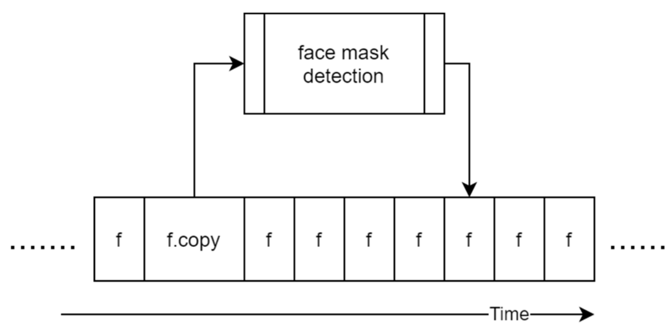

Figure 8. The buffer copying delay

To avoid the frame-drop from the detection algorithm, we used the multithreading technique. Because the detection algorithm can cause the delay that leads to the frame-drop, it is assigned to a second thread. As shown in Figure 7, the detection algorithm no longer blocks the video frames when it runs in the second thread.

However, a video frame cannot be pushed into the face mask detection algorithm immediately. Another significant delay comes from copying the current video frame to a buffer, as shown in Figure 8. Copying frame to buffer is required for freezing the frame and failing to do so results in the buffer being changed while still waiting for processing in the face mask detection algorithm. The delay cannot be avoided by multithreading as the video buffer needs to stay in the current frame until the copy procedure finishes. In our test, the buffer copying brings about a 47.2ms delay on average. The frame rate thus can be estimated to be

$$f = 60\left(1 - \frac{t_{copy}}{t_{copy} + t_{detection} + t_{frame}}\right) \approx 41\text{fps} \qquad (3)$$

This frame rate can be improved by adding a small delay between each detection, as increasing the detection time will increase the denominator in equation (3), though at the cost of an increased detection delay.

III. Experiment Results and Evaluation

To fine-tune the face mask detection system, the *Medical mask dataset* [11] is used for tuning the configurations of the HCCs used in our face mask detection algorithm. This dataset contains a total of 6024 images of people with or without face masks.

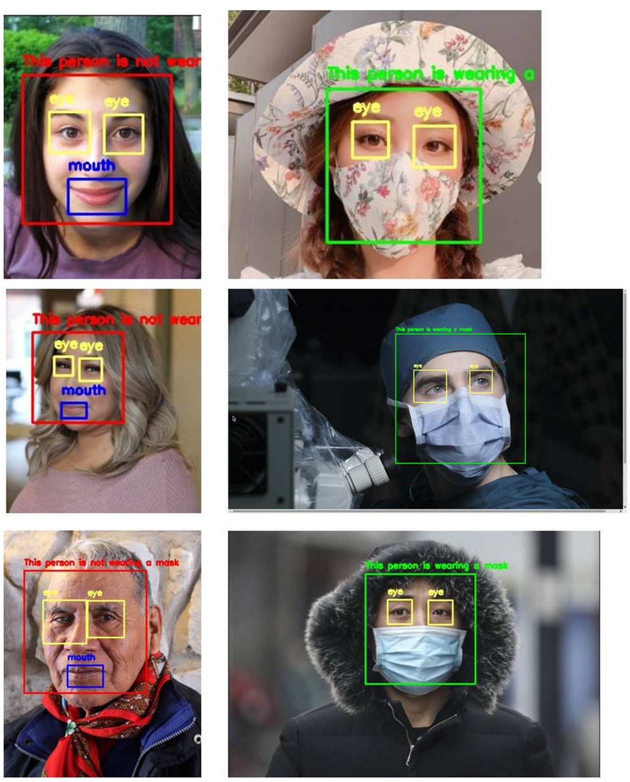

Figure 9. A selection of test images with detection results

A subset of 100 images is used for tuning our algorithm, and Figure 9 shows a small selection of test images with the detection results, where green rectangles mean the person in the image is wearing a mask. Note that eye detection is also included here but is discarded in the real-time face mask detection system for better efficiency. Using the test images, we can determine the configurations for the HCCs that suit our algorithm the best. For the face detection, the scaling factor is set to 1.3 and the minimal neighbor is set to 5. For the mouth detection, the scaling factor is set to 1.1 and the minimal neighbor is set to 4. Those configurations allow for a good face detection rate and sensitive mouth detection.

The face mask detection system is then tested in the following scenario: the test subject sits in front of the camera with a face mask on for 10 seconds and then takes off the face mask for 10 seconds and then repeats. The camera footage is then streamed into the HDMI input port on the PYNQ board. The output of the PYNQ board is connected to a monitor to visualize the detection output.

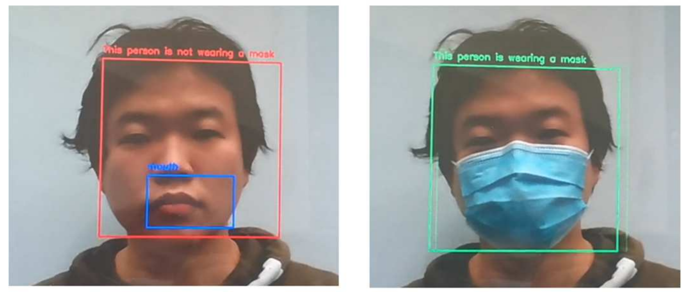

Figure 10. Successfully detecting the mask-wearing status

Table I. The test results

Results	times
Total detections	354
Mask on	162
Mask off	192
False negative	0
False positive	7
Offset	100%
Precision	96.5%

Figure 10 shows the detection system successfully detecting the subject's mask-wearing status. The test ran for 46 seconds, and as shown in Table I, 354 detections took place in this period. In those detections, 192 frames of inputs have the ground truth of mask off, so the test subject was not wearing a face mask in those frames. For the other 162 frames, the test subject was wearing a face mask. We assign the mask off status as the positive because it is more important to detect a person not wearing a mask and make an alert. In the test, all of the mask off frames are detected positive, yielding an offset of 100%. On the other hand, seven frames where the test subject wearing a face mask are also detected positive. As those are false positives, the precision is 96.5%. The input is at 60fps, 720P, while the average frame rate of the output is measured to be 45.79fps on average at the same resolution. This result roughly matches the estimation from equation (3). When calculating the average run time of the algorithm, the detection delay is $83.1ms$ from a total of 170 detections. In addition, the buffer copying delay should also be added to the total delay, which yields a total detection delay of $130.3ms$, or $0.13s$.

IV. Conclusion

In this study, a face mask detection system built on the PYNQ SoC platform is presented. The system can detect the mask-wearing status of persons showing up in a camera feed or video stream in general, and mark the detection result on the video output. Haar cascade classifiers are used for the face mask detection algorithm. Different delays are analyzed in an attempt to reduce the delays in the system to achieve real-time detection. In the end, the detection system is demonstrated to be able to run at 45.79fps in 720P, with a $0.13s$ detection delay and a detection precision of 96.5%.

For future works, the face and facial detection algorithm can be modified to use CNN or other deep learning techniques for a better detection rate. The detection algorithm can also be implemented in the programmable logic array for lower detection delay.

References

[1] B. Rader, L. F. White, M. R. Burns, J. Chen, J. Brilliant, J. Cohen, J. Shaman, L. Brilliant, M. U. G. Kraemer, J. B. Hawkins, S. V. Scarpino, C. M. Astley and J. S. Brownstein, "Mask-wearing and control of SARS-CoV-2 transmission in the USA: a cross-sectional study," *The Lancet Digital Health,* vol. 3, pp. e148-e157, 2021.

[2] P. Viola and M. Jones, "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 2001*, 2001.

[3] L. Cuimei, Q. Zhiliang, J. Nan and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," in *2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, 2017.

[4] A. Kasinski and A. Schmidt, "The architecture and performance of the face and eyes detection system based on the Haar cascade classifiers," *Pattern Analysis and Applications,* vol. 13, p. 197–211, 3 2009.

[5] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector," 2020.

[6] M. Loey, G. Manogaran, M. H. N. Taha and N. E. M. Khalifa, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sustainable Cities and Society,* vol. 65, p. 102600, 2021.

[7] T. Q. Vinh and N. T. N. Anh, "Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier," in *2020 International Conference on Advanced Computing and Applications (ACOMP)*, 2020.

[8] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustainable Cities and Society,* vol. 66, p. 102692, 2021.

[9] M. S. Ejaz and M. R. Islam, "Masked Face Recognition Using Convolutional Neural Network," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2019.

[10] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*, Berlin, 1995.

[11] "Medical mask dataset," Humans in the Loop, 2020.