Abstract: Mask-face detection has been a significant task since the outbreak of the COVID-19 pandemic in early 2020. While various reviews on mask-face detection techniques up to 2021 are available, little has been reviewed on the distinction between two-class (i.e., wearing mask and without mask) and three-class masking, which includes an additional incorrect-mask-wearing class. Moreover, no formal review has been conducted on the techniques of implementing mask detection models in hardware systems or mobile devices. The objectives of this paper are three-fold. First, we aimed to provide an up-to-date review of recent mask-face detection research in both two-class cases and three-class cases, next, to fill the gap left by existing reviews by providing a formal review of mask-face detection hardware systems; and to propose a new framework named Out-of-distribution Mask (OOD-Mask) to perform the three-class detection task using only two-class training data. This was achieved by treating the incorrect-mask-wearing scenario as an anomaly, leading to reasonable performance in the absence of training data of the third class.

Keywords: COVID-19; face mask; out-of-distribution; deep learning; machine learning

1. Introduction

COVID-19 has posed great challenges to the public globally since the first case of unknown pneumonia detected in humans in December, 2019. On 30 January 2020, the World Health Organization (WHO) announced that the epidemic of COVID-19 was listed as a public health emergency of international concern (PHEIC). In March 2020, the World Health Organization further announced that the outbreak of COVID-19 constituted a global pandemic. By 10 July 2022, 561 million cases of COVID-19 had been confirmed, including over 6 million deaths, according to WHO’s report [1]. During the continuous outbreak of the COVID-19 epidemic, various mutants such as Delta and Omicron emerged, which led to an explosive increase in the confirmed cases, requiring effective epidemic prevention and control measures. With the rise of new and more transmissible sub-variants of BA.4 and BA.5 [2], it seems the COVID-19 pandemic is far from over yet.

Wearing face masks is an important measure to prevent the transmission of diseases among people. Hence, face mask wearing in public places is still encouraged or even mandatory [3]. To ensure the effectiveness of face masks, according to the guidelines of WHO, one should wear face masks correctly (i.e., making sure the nose, mouth and chin are all covered). Therefore, some countries in the epidemic established monitoring systems to guide people to put on face masks properly in public places, such as public transportation, shopping malls, hospitals, etc. Meanwhile, many researchers have worked on effective face mask detection methods and hardware systems.

With the large amount of papers dealing with face-mask detection methods and hardware systems, it is natural to take stock to effectively review these papers for the adoption
of those more efficient face-mask detection methods and hardware systems in practical use. While several review papers on this topic have been published in the past years [4–7], a common limitation of these review papers is that they all mainly concentrate on reviewing the methodologies and software developed, i.e., without reviewing face-mask detection hardware systems properly and systematically. Moreover, while incorrect mask-wearing has been touched upon by some of the review papers [4,5], no in-depth review is available as of yet. With this background, in this paper, we review face-mask detection methods by singling out both the two-class and three-class cases, where two-class refers to wearing a mask and without mask and three-class has an extra class of incorrect mask-wearing. The main gap between the two cases is analyzed. Moreover, we reviewed papers on face-mask detection hardware systems specifically. In addition, a new framework is introduced to handle the incorrect mask-wearing detection task in the absence of training data of the third class. The main contributions of the study are as follows:

- We present an up-to-date review of recent face-mask detection methodologies in both two-class and three-class cases and investigate the gap in techniques between them.
- We provide an additional review and analysis on mask-face detection hardware systems. Various techniques of implementing mask-face detection models into portable systems are also considered.
- We develop a new Out-of-distribution-Mask (OOD-Mask) detection framework, which can perform three-class mask-face detection with only two-class data of wearing a mask and without mask.

The outline of the remainder of this paper is as follows: We first explain our review strategy in Section 2. Then, we make a brief summary in Section 3 of commonly used evaluation criteria and previous review papers on mask-face detection methodology in recent years. Following that, we provide a review of face-mask detection methodologies in terms of both two-class and three-class features in Section 4, as well as a review of face-mask detection hardware systems in Section 5. In Section 6, we introduce the new OOD-Mask detection framework, including its implementations and experiments. Finally, we conclude our work and present several research directions for future works for face-mask detection in Section 7.

2. Reviewing Methods

In this study, the review was conducted following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines. We divided the filtering process into the following several phases.

2.1. Searching Strategy

We first clarified our research objectives of an up-to-date review for two-class and three-class mask-face detection models and a review for mask-face detection systems. Therefore, we explored online electronic databases including Google Scholar, IEEE Xplore Digital Library, Web of Science, and Springer Link with the following Boolean search strings:

- (“face mask” OR “mask face”) AND (“detection” OR “detection model”).
- (“face mask” OR “mask face”) AND (“detection system” OR “detection hardware” OR “detection hardware system”).

2.2. Study Selection

We collected a total of 2569 articles from the searched databases as shown in Figure 1. We first excluded all duplicated articles by different Boolean search strings. Then, we made a further screening according to articles’ titles and abstracts manually.
Figure 1. PRISMA diagram for two-class mask-face detection, three-class detection and mask-face hardware systems. * December 2019 refers to the time of the COVID-19 outbreak.

2.3. Eligibility Check

After the relevance selection, we further evaluated the eligibility for the remaining 135 articles. For the up-to-date reviews of mask-face detection models, we excluded duplicated articles from previous reviews [4,5]. For the review of mask-face detection hardware systems, we excluded articles without model descriptions or hardware specifications.

Consequently, we retained 48 articles for our review purposes in this study. These 48 articles consist of two parts. First, 32 represent the latest research on mask-face detection models which are not mentioned in previous reviews and are analyzed in Section 4 to show the state-of-the-art methods for improving mask-face detection performance. The other 16 articles covered most mask-face detection systems based on various research angles according to the review in Section 5. Thus, the articles we collected are sufficient to achieve our review aims.

2.4. Review Structure

Additionally, to make our review clearer and understandable, we visualized the overall review structure as in Figure 2. Finer classification details are provided in each review part and we explain them in detail in the corresponding Sections.
Figure 2. The overall mask-face detection review structure.

3. Related Works

3.1. Major Evaluation Metrics

Due to different characteristics of the approaches taken in the literature, studies on mask-face detection methodologies use a variety of indicators to evaluate their performance. In this section, we introduce these evaluation metrics used frequently in our investigated research, and the corresponding formulas.

We first provide the definitions of the required statistics. True Positive ($T_P$) indicates that the positive sample is predicted correctly. False Negative ($F_N$) refers to the positive sample being incorrectly predicted to be of negative class. True Negative ($T_N$) defines cases of a negative sample being correctly classified into the negative class. Additionally, False Positives ($F_P$) are the result of negatives samples incorrectly classified into the positive class.

Recall that Equation (1) indicates how many positive samples are correctly detected in the prediction result. The precision shown in Equation (2) is defined as evaluating the proportion of correctly predicted samples. Since the values of precision and recall are inversely proportional to each other, sometimes there are contradictions. Thus, it is necessary to use F1-score to weight and average these two indicators, as expressed in Equation (3).

\[
Recall = TPR = \frac{T_P}{T_P + F_N} \quad (1)
\]
\[
Precision = \frac{T_P}{T_P + F_P} \quad (2)
\]
\[
F1-score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)
\]

The True Positive Rate ($TPR$) has the same definition as recall (1). The False Positive Rate ($FPR$) shown in Equation (4) refers to the proportion of wrongly predicted negative samples with respect to all negative samples. $TPR$ and $FPR$ are often used as a pair of evaluation indicators. Accuracy, expressed in Equation (5), is the most common indicator which indicates the proportion of correctly classified samples with respect to the overall sample space. In general, a higher accuracy suggests better classification performance.

\[
FPR = \frac{F_P}{T_N + F_P} \quad (4)
\]
\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}
\] (5)

In order to measure the overlap degree between the predicted bounding box and the ground truth, the Intersection over Union (IoU) was utilized as shown in Equation (6). By adjusting the value of IoU, a corresponding set of recall and precision values can be obtained. The Average Precision (AP), expressed in Equation (7), is the area enclosed by this set of values. The mean Average Precision (mAP) is calculated by averaging all AP values of N categories as defined in Equation (8).

\[
\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}
\] (6)

\[
AP = \int_{0}^{1} p(r)dr
\] (7)

\[
mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i
\] (8)

3.2. Summary of Previous Review Papers on Mask-Face Detection Methodologies

Research on mask-face detection emerged in recent years in varied datasets, model types, evaluation criteria and applied systems. This motivated several researchers to review and summarize different approaches to establish reliable guidelines for future development. In this section, we discuss in detail and summarize these previous literature reviews on mask-face detection.

Since mask-face detection is an application of object detection algorithms, Nowrin et al. [4] first divided mask-face detection models into machine learning-based algorithms and deep learning-based algorithms, then divided each category of models further. Traditional methods were divided into Viola–Jones detectors [8], HOG-based detectors, etc. Methods based on deep learning were classified as CNN, GAN and other deep neural networks. Additionally, the reviews stressed the importance of the remaining image resolution during detection, the scarcity of diverse datasets and classification of objects, incorporated issues such as the diversity of datasets and mask types, different mask wearing conditions, and reconstruction of masked faces for future improvements.

In the review provided in [5], the authors introduced state-of-the-art neural network models and datasets used in mask-face detection comprehensively and evaluated the prediction accuracy. They also detailed the shortcomings of different models. As the conclusion, this review proposed future research directions for detecting mask-wearing conditions in real scenarios; highlighted the demand for a shared mask database to develop a high-accuracy and real-time automatic mask-face detection system; and proposed the fitting of a deeper and wider range of deep learning architectures of more training parameters for the mask-face detection task, such as Mask R-CNN [9], Inception-v4 [10], Xception [11], etc.

In another study [6], face-mask detection models were categorized into one-stage detection, two-stage detection, and feature extraction. The study summarized and analyzed the real-time and mask-face detection technologies and compared the advantages and challenges among different structures. After reviewing all the research, it was found that the utilization rate of deep learning surged after the COVID-19 outbreak. In addition, the reviews proposed that the quality of datasets can be improved by removing images with insufficient light, and by combining different detection models with systems that can detect whether a sufficient physical distance is maintained between people.

In [7], Ali et al. introduced several techniques of mask-face detection and briefly analyzed their drawbacks according to multi-stage and single-stage detection. They divided multi-stage detection models into CNN, R-CNN [12], Fast R-CNN [13], Faster R-CNN [14], and Mask R-CNN [9]. Similarly, YOLO [15] was introduced as single-stage detection.
However, the review mainly discussed the architectures of general object detection models, rather than focusing on the characteristics of mask-face detection itself.

In the above literature reviews, the researchers all first extended face-mask detection to a general object detection task. By collecting the reported datasets, classifying and analyzing different model architectures, studies put forward multiple prospects for improving mask-face detection models in the continuous epidemic. Nevertheless, these studies shared the limitation of paying little attention to distinguishing the mask-face categories and failing to focus on the techniques of detecting the third class of incorrect mask-wearing. Moreover, none of the previous reviews thoroughly explored the application of mask-face detection models to hardware systems.

To fill the gaps left by the above mentioned reviews, we divide the research in mask-face detection into two-class and three-class reviews in the next section and provide another mask-face detection hardware system review in Section 5. In each review, more detailed categorizations are made according to the corresponding criteria.

4. Review of Mask-Face Detection Methodologies

In this section, we summarize various studies focused on mask-face detection published in recent years. We first collate commonly used datasets for mask-face detection in Table 1.

Table 1. Common datasets for mask-face detection.

| Dataset            | Year | Images (or Videos) | Class                              |
|--------------------|------|--------------------|------------------------------------|
| WIDER FACE [16]    | 2016 | 32,203 images      | without mask                       |
| MAFA [17]          | 2017 | 30,811 images      | wearing mask                       |
| MFDD [18]          | 2020 | 24,771 images      | wearing mask                       |
| SMFRD [18]         | 2020 | 500,000 images     | wearing mask                       |
| RMFRD [18]         | 2020 | 95,000 images      | wearing mask: 5000/without mask: 90,000 |
| Moxa3K [19]        | 2020 | 3000 images        | without mask: 9161/wearing mask: 3015 |
| AIZOO [20]         | 2021 | 7971 images        | wearing mask: 4064/without mask: 3894 |
| FMD [21]           | 2021 | 7553 images        | wearing mask: 3725/without mask: 3828 |
| MAFA-FMD [22]      | 2021 | 56,024 images      | wearing mask: 28,233/without mask: 26,463/incorrect mask-wearing: 1388 |
| WMD [23]           | 2021 | 7804 images        | wearing mask: 26,403               |
| ViDMASK [24]       | 2022 | 67 videos          | wearing mask: 26,403 instances/without mask: 2500 instances |

4.1. Two-Class Mask-Face Detection Methodologies

Due to the difficulty of collecting and labeling the class of incorrect mask-wearing, the majority of research focused on two-class (i.e., wearing mask and without mask) mask-face detection. In order to achieve the real-time detection of masks during the epidemic, in addition to the popular deep learning methods used for mask-face detection, there are a small number of studies that used traditional machine learning methods to extract and classify object’s features. In this part, we review the research on mask-face detection from both traditional machine learning and deep learning methods. Table 2 shows the discussion and classification of related research works.
| Ref. & Publish Date | Datasets | Models | Detection Performance | Merits | Limitations |
|--------------------|---------|--------|-----------------------|--------|-------------|
| October 2020, [25] | A combination of CASPEAL dataset, AISL dataset, INRIA dataset and RMFD dataset | Haar Wavelet, Local Binary Pattern (LBP) and Histogram of Oriented Gradient (HOG) based Adaptive Boostings and Cascade Classifiers | Accuracy of 86.9% | The work focused on masks multi-pose faces using traditional methods. | The detection accuracy was lower than that of the neural network, and self-made feature pictures were limited. |
| January 2021, [26] | A customized dataset of 2192 images | FMRN (A customized CNN) | Day time accuracy of 95.8%, Night time accuracy of 94.6% | The authors added the human skeleton analysis part to analyze the human posture, and the influence of light on the detection system was considered. | The number of images in the dataset was limited. |
| July 2021, [27] | A customized mask-face dataset and video surveillance | ResNet50 [28], MobileNet [29] and AlexNet [30] | Accuracy of 98.2% | Besides mask-face detection, the identity of persons without a mask was confirmed. It required less memory and has low inference time, high accuracy. | Due to the addition of RPN, the detection speed was not as fast as one-stage. |
| March 2021, [31] | A customized dataset of 4095 images | A transfer learning on MobileNetV2 [32] and use SSD [33] detector. | Accuracy of 98% | Performed high detection accuracy and could work on devices with limited computational resources. | The proposed method lacked consideration for crowded situations. |
| June 2021, [34] | The customized dataset of 350 MB size | Mask Detection Model (MDM) with YOLOv3 [35] detector | Average confidence level are 97% (A), 96% (B) and 93% (C) | Separated the data categories into individual images (type A), group people (type B), and videos with the group of people (type C). | The study lacked comparison with other mask-face detection models. |
| December 2021, [36] | A combination of 1916 masked and 1930 unmasked images | Stacked ResNet50 [28], YOLOv5 and density-based spatial clustering of applications with noise (DBSCAN) | mAP of 98% | In addition to detecting masks, social distancing norms were also detected, and real-time detection was performed. | Dim light situations were not investigated in the scope of the research. |
### Table 2. Cont.

| Ref. & Publish Date | Datasets | Models | Detection Performance | Merits | Limitations |
|---------------------|----------|--------|------------------------|--------|-------------|
| January 2022, [37]  | A customized dataset of 9000 images | YOLOv5 | Accuracy of 92% | The authors combined attention mechanism with YOLOv5, thereby improving the accuracy of mask wearing detection in dim light conditions. | The dataset did not contain pictures of multiple people for mask-face detection. |
| February 2022, [38] | A customized dataset of 4000 images | A customized CNN model with ResNet-10 [28] architecture based on SSD | Accuracy of 98% | Compared with other network models, the accuracy rate was higher for a small model size. | This model did not consider mask-face detection in dim light situations. |
| February 2022, [39] | WiderFace and MAFA | A customized frame “CenterFace” (CenterFace(MobileNet) (CenterFace(ResNet18) and an anchor-free detector | Without mask: precision of 99.3%, recall of 96.2%, wearing mask: precision of 99.7%, recall of 96.7% | They conducted ablation studies using different datasets and networks, and the size of proposed model was small. | Surveillance video was not included in the dataset, and real-time detection was not considered. |
| February 2021, [40] | A combination of MAFA, WIDER FACE and many manually prepared images | Faster R-CNN [14] and YOLOv3 [35] | Average Precision of 55% and 62% | Comparing one-stage and two-stage models in mask detection for the first time, it was found that two-stage had higher accuracy and that the one-stage model is faster. | There was a limited contribution of deep learning models to mask-face detection. |
| May 2022, [24]     | ViDMASK (proposed for the first time) | Mask R-CNN [9], YOLOv4 [41], YOLOv5 and YOLOR [42] | mAP of 92.4% in YOLOR | The research proposed a new dataset, ViDMASK, in terms of pose, environment, quality of image, and versatile subject characteristics. | Compared with MOXA3K, except for the better performance on YOLOR, it did not perform well on the rest of the frameworks. |
| April 2021, [43]   | SMFD     | Customized CNN and VGG16 [44] based deep learning models | Accuracy of 97.42% and accuracy of 98.97% in test | The model added data augmentation to enlarge the training dataset. | A limited number of images were included in the dataset, and the model did not detect during video surveillance. |
Table 2. Cont.

| Ref. & Publish Date | Datasets            | Models                                      | Detection Performance | Merits                                      | Limitations                                      |
|---------------------|---------------------|---------------------------------------------|-----------------------|---------------------------------------------|--------------------------------------------------|
| May 2021, [45]      | MFRD                | A customized DenseNet201 model              | F-Measure of 98%      | It had higher detection accuracy with more parameters. | Only the front face was detected, and different face poses cannot be detected. |
| March 2022, [46]    | FMD, FM, RMFRD      | MobileNetV2 [32] followed by autoencoders.  | precision of 99.96%, recall of 99.97%, F1-score of 99.97%, and accuracy of 99.98% | Ablation studies were conducted in nine network structures using three different datasets and the proposed models was compared with other models. The proposed model size is small. | The model did not provide location information, as it only focused on the classification. |
| March 2022, [47]    | RMFD                | A customized deep CNN                       | Accuracy of 99.57%    | The model achieved high accuracy.            | The dataset did not include scenes with many people. |
| March 2022, [48]    | A customized mask-face dataset | SSDMNV2, LeNet-5, AlexNet [30], VGG-16 [44], ResNet-50 [28] and SVM | Accuracy of 98.7%    | They compared multiple models to find the algorithm of best accuracy while spending the least time during training and detection. | The proposed method was not described in detail and the results of performance were not clear for comparison. |
| May 2022, [49]      | A customized mask-face dataset | HIC-DEEP, using InceptionV3 [50] and evaluation of the distance algorithms (Euclidian, Spearman, and Pearson) | Accuracy of 60%, 78% and 85% | The architecture chose three different distance measure methods to classify images. | The accuracy of the proposed model was low. |
4.1.1. Traditional Machine Learning Based Approaches

Before deep learning methods became popular, researchers tended to use traditional machine learning for feature extraction. The location and range of the region of interest are determined by letting candidate frames of different sizes traverse the entire image, and then feature extraction algorithms are used such as Haar and Histogram of Oriented Gradient (HOG) to extract features from the candidate region, so as to obtain the required features and classify them.

In the study of Dewantara et al. [25], their feature-based adaptive boosting and cascade classifiers trained Haar-like features, Local Binary Pattern (LBP), and HOG, respectively, to obtain the required features and perform sorting. Then, since Haar-like features focused more on facial contours (nose and mouth), its accuracy reached 86.9%, surpassing the 81.7% accuracy of LBP and 74.5% of HOG.

4.1.2. Deep Learning Based Approaches

In the field of machine learning, deep learning-based methods have achieved great popularity in recent years. Deep learning aims to build and simulate the neural network of the human brain for analysis and learning. It imitates the mechanism of the human brain to interpret data such as images, sounds, and texts. Many architectures have been proposed in object detection with the development of deep learning. As a branch of object detection, mask-face detection also includes the processes of target detection, feature extraction, classification, etc. According to the implementation of object detection methods, we divided mask-face detection models into the following two categories: two-stage and one-stage detection methods. In addition, there are some studies that only performed classification using convolutional neural networks and learned nothing about location information. Therefore, we divided deep learning methods into the following three parts: two-stage detection, one-stage detection and classification.

A. Two-stage detectors

Due to the lack of human and material resources and incomplete feature coverage in previous traditional machine learning models, two-stage detection generates a series of sample candidate boxes using the RPN [14] algorithm to obtain a more accurate detection effect. The main network includes the R-CNN series [13,14,51], etc.

Sethi et al. [23] proposed a novel two-stage detector that also possessed the advantages of one-stage detectors. The model achieved a low inference time and high accuracy of 98.2%. In addition, adding affine transformation could more accurately locate faces in different poses so as to identify people who are not wearing masks and retrieve their identity information for registration. This method was helpful for shopping malls, schools, and other public places to check whether masks are worn through surveillance video.

B. One-stage detectors

Compared with the high accuracy of two-stage detectors, more research works tend to use the one-stage detector to detect and extract object features in pictures or videos with a higher detection efficiency, which could help achieve the target of real-time mask-face detection. One-stage detection methods mainly include the You Only Look Once (YOLO) series [15,35,52], Single Shot Detector (SSD) [33], etc. They directly returned the category probability and position coordinate values of objects through the model. Such an algorithm can provide faster detection, but the accuracy may be lower than that of the two-stage detection network.

In the work of Guo et al. [37], in order to solve the problem of the low accuracy of mask-wearing detection under dim light conditions, a method of introducing an attention mechanism into the YOLOv5 network for mask-wearing detection was proposed. By using CIoU Loss [53] to calculate the localization loss of the target frame, it not only considered the overlapping area of the predicted frame and the real frame but also accounted for the distance between the center points of the two and the aspect ratio of the two. At the same time, a convolutional block attention module (CBAM) [54] was added to the YOLOv5
network, which multiplied the input feature map and the attention map for adaptive feature optimization, highlighting the main features and suppressing irrelevant ones. The accuracy of this model reached 92% in the detection of mask-wearing under dim light conditions.

C. Classification

In some studies, the input images were preprocessed with methods such as enlarging, cropping, etc., to maintain only the focused areas for classification. They only focused on classification architectures, leaving out the detection component.

In order to improve the accuracy of mask-face classification, Pranjali et al. [48] proposed a new idea to focus on the parts of non-covered faces and obtained the best aspects of the collection areas through CNN. They made a comparison among different classification models of SSDMNV2, LeNet-5, AlexNet [30], VGG-16 [44] and ResNet-50 [28]. As a result, their system achieved 98.7% accuracy.

In [46], Habib et al. first performed data augmentation on the following three datasets: Face Mask Detection (FMD) [21], Face Mask (FM) [55], and Real-World Mask Face Recognition (RMFR) by implementing flipping, rotation, etc., to expand the size of training data. After that, they proposed a model based on the MobileNetV2 [32] architecture, followed by autoencoders, which were used to transform the high-dimensional output feature vector into low dimensions. Finally, they performed ablation studies with their proposed model with VGG16, EfficientNet [56] and other well-known CNN models. Their model achieved a precision of 99.69%, recall of 99.97%, F1-score of 99.97%, and accuracy of 99.98%. Furthermore, the light-weighted architecture of the proposed model achieved 199.01 FPS over GPU, 44.06 FPS over CPU, and 18.07 FPS over the Raspberry Pi resource-constrained device.

Besides the two-class mask-face detection methods, some researchers paid attention to the issue of incorrect mask-wearing, which plays a critical role in the protection of infection during the epidemic. Therefore, in the next section, we summarize studies on mask-face detection models of three-class, which includes wearing mask, without mask, and incorrect mask-wearing.

4.2. Three-Class Mask-Face Detection Methodologies

The main defect of two-class based methods is that in the case of incorrect mask-wearing (i.e., wearing a mask without covering the nose, mouth or chin), the mask can still be detected in the image but it is not effective in virus prevention. Therefore, researchers recently paid more attention to performing mask-face detection with the additional class of incorrect mask-wearing.

The detection categories used for three-class mask detection are roughly the same as the two-class methods. Due to the characteristics of three-class datasets, three-class mask detection is more challenging.

Firstly, a majority of datasets used by existing three-class detection methodologies are unbalanced and the number of samples in the incorrect mask-wearing class is less than that in the class of wearing mask or the class of without mask. In addition, the feature of incorrect mask-wearing has a high similarity with the other two categories, and this directly leads to the lower accuracy of the class.

In order to improve the accuracy of detecting incorrect mask-wearing, some studies [57–60] added attention mechanisms to balance features from multiple dimensions to generate more salient features. Other studies [61,62] searched for and labelled more incorrect mask-wearing data by themselves or used the data augmentation approaches to increase the small number of samples in the incorrect mask-wearing class, thus making the three classes in the training dataset more balanced as to achieve a better mask-face detection performance. Studies used for three-class mask-face detection are discussed in Table 3 after introducing representative studies from each category in the subsequent sections.
Table 3. Summary of three-class mask-face detection methodologies.

| Ref. & Publish Date | Datasets | Models | Detection Performance | Merits | Limitations |
|--------------------|----------|--------|------------------------|--------|-------------|
| **Attention-based approach** | | | | | |
| March 2021, [60] | A customized dataset of 4672 images | A customized framework of Context-Attention R-CNN | mAP of 84.1% | The model improved the accuracy of mask-face detection by adding an attention module, decoupling branches, and adapting multiple extractors of the context features. | Three classes of data in the dataset were imbalanced, without a discussion of the model size. |
| November 2021, [57] | Two publicly available datasets | A customized network FMD-Yolo | AP50 of 92.0% and 88.4% on the two datasets | Improved robustness, generalization ability and inference efficiency. | The accuracy for the incorrect mask-wearing class was lower than for the other two classes, and the model did not include the impact of different scenarios on mask-face detection. |
| May 2022, [58] | A customized dataset of 11,000 images | SMD-YOLO based on YOLOv4-tiny [41] | mAP of 67.01% | The model significantly improved the detection ability of small targets in public, and had a fast processing speed for real-time detection. | The data imbalance effect was not considered. |
| June 2022, [59] | A combination of RMFD, AFLW, and a web crawler | YOLOv5-CBD | mAP of 96.7% | This model reached high accuracy with potential for real-time detection. It was able to detect small-scale targets, which led to better effectiveness compared to YOLOv5. | There was only an overall performance of three classes provided so we were unable to evaluate each class separately. |
| **Data balancing-based approach** | | | | | |
| March 2021, [23] | Proposed models of WMD and WMC | Based on the Faster R-CNN [14] and InceptionV2 [63] in the first stage. The face masks were verified by a broad learning system. | Correct Mask accuracy of 99.87% Incorrect Mask, Chin: Accuracy of 99.93% Incorrect Mask, Mouth, Chin accuracy of 97.47% | They considered both the impact on mask detection in different scenarios and situations where multiple people wear masks in small-size images. | Real-time detection in video surveillance was not tested. |
### Table 3. Cont.

| Ref. & Publish Date | Datasets | Models | Detection Performance | Merits | Limitations |
|---------------------|----------|--------|------------------------|--------|-------------|
| May 2021, [62]      | A combination of their own photos, RMFD and MaskedFace-Net datasets | The combination of an improved CSPDarkNet53, the adaptive image scaling algorithm and the improved PANet structure based on YOLOv4 [41] | mAP of 98.3%, when IOU was 0.5. | The developed model had a high accuracy and high frame rate. The customized dataset included various head poses wearing masks and various situations of wearing incorrect masks. | This study performed mask-face detection without the consideration of dim light conditions. |
| August 2021, [61]   | A customized dataset of 7892 images | Face detection algorithms in OpenCV & ResNet50 [28] and NASNetMobile [64] used for feature extraction and classification | Accuracy of 92% | Made a comparison between light-weighted network and deep structure network. The numbers of samples of each class in the dataset are about the same. | Neglected the situations of mask-face detection in video surveillance. |
| October 2021, [65]  | A customized dataset of 3300 images and four videos | Faster R-CNN [14] and Inception V2 [63] | Accuracy of 99.8% | The algorithm successfully identified people who did not maintain physical distance and were too near each another. It also took into account the angle of the camera. | Compared with the other two types of results, the accuracy of incorrect mask-wearing class was not satisfactory. |
| March 2021, [66]    | A customized dataset of 52,635 images | Tweaked and modified feature extraction networks for tiny YOLOv1 [15], tiny YOLOv2 [52], tiny YOLOv3 [35] and tiny YOLOv4 [41] | mAP of 70.25% for M-T YOLOv4 | The proposed dataset contained a unique class of mask area, which enhanced the detection accuracy of small object targets. | There were large gaps between the accuracy of the three classes of detection, especially for incorrect mask-wearing. |
| Others              | A combination of 14,409 images from Wider face, FMD and RMFD | Yolov4FaceMask | mAP of 88.82% | The research considered the detection of different types of face masks in different scenarios and input videos of different resolutions. | The detection accuracy was low. The number of samples in each class in training dataset was not specified. |
Table 3. Cont.

| Ref. & Publish Date | Datasets | Models                                      | Detection Performance | Merits | Limitations                                                                 |
|---------------------|----------|---------------------------------------------|-----------------------|--------|-----------------------------------------------------------------------------|
| August 2021, [68]   | A customized dataset of 3200 images | CNN Models of MobileNetV2 [32], Xception [11], InceptionV3 [50], ResNet-50 [28], NASNet [64], and VGG19 [44] with transfer learning. | F1-score of 85% | In the dataset, the class of incorrect mask-wearing was subdivided into ten more specific types. | The detection accuracy of the incorrect mask-wearing was low. |
| October 2021, [69]  | A customized dataset of 11,000 images | Tiny YOLO v4-SPP | mAP of 64.31% | In addition to the detection of the classes of masked-faces, the mask area was also detected. | The detection accuracy was lower than other state-of-the-art methodologies, and the accuracy gap among categories was large. |
| December 2021, [70] | ISL-UFMD and ISL-UFHD (proposed for the first time) | RetinaFace as the detector. ResNet50 [71], Inception-v3 [30], MobileNetV2 [32] and EfficientNet [56] as classifier. And the DeepHRNet [72] for person detection. | Accuracy of 98.20% | The authors not only considered mask-face detection, but also the detection of face–hand interaction and social distance. | Compared with the other two categories, the class of incorrect mask-wearing was of a low accuracy. |
4.2.1. Attention-Based Approaches

The attention mechanism is a component used in neural networks to model long-range interaction, which has led to great progress in both NLP and CV fields.

In Computer Vision, motivated by human observation, the attention mechanism is a dynamic weight adjustment process based on features of the input images. In the study of three-class mask-face detection [57–60], researchers added the attention mechanism after extracting the target features through the neural network, with the aim of reducing the effect of the class imbalanced problem on the detection performance.

In particular, Zhang et al. [60] proposed a two-stage structure with the attention mechanism, Context-Attention R-CNN, which could reach mAP of 84.1%. The dataset contained 4672 images in three categories, including 3988 images without masks, 636 images wearing masks, and 48 images of incorrect mask-wearing. Among them, the incorrect mask-wearing category had the least number of images, presenting a large gap with the other two categories. In order to reduce the effect of this imbalance on the detection performance, the attention mechanism was added after extracting the target features through the neural network. It could weigh the extracted features to obtain more effective features of masked-faces from different dimensions and reduce noise features. The mAP of incorrect mask-wearing was improved by 5.5 percent using this method.

4.2.2. Data Balancing-Based Approaches

Due to the imbalanced problem in the existing dataset, [23,61,62,65] established datasets of three balanced categories by making more annotations for incorrect manually. More specifically, in [65], by performing enhanced data augmentation techniques, the training dataset represented an equal proportion of the three categories. In addition, their model was more robust in real scenarios owing to data augmentation. Their Faster R-CNN Inception ResNet V2 model finally achieved an accuracy of 99.8%.

Kayali et al. [61] also customized their dataset with equal amounts of data in three classes before designing their mask-face detection model. This dataset avoided the situation where three classes of detection results were quite different due to the imbalance of training data. First, they used the face recognition algorithm in OpenCV to locate and crop the faces in the images, and the obtained pictures passed through the lightweight network NASNetMobile [64] and the deep-structured ResNet50 [28], respectively. After comparison, the model achieved an accuracy of 92%.

4.2.3. Others

Other studies did not design mask-face detection methodologies from the perspective of attention mechanism or data imbalance. Instead, they made contributions from other perspectives. In the research of of [66,69], they introduced a new class of mask-face detection named mask area, which provided the location information of masks. It could also served as a potential method for generating images of mask-covered faces. For simplicity, we attributed them to three-class mask-face detection methodologies since their main purposes were to detect incorrect mask-wearing. However, whether such classification would be useful in practical mask-face detection remains to be examined. On the other hand, their proposed methods presented large gaps in terms of the accuracy of different classes.

Besides detecting masked faces, Eyiokur et al. [70] paid attention to the social distance and the interaction between face and hand. The attention model was leveraged for the detection of face–hand interaction. Combining such additional components helped to track and prevent infection with COVID-19 more comprehensively.

The papers mentioned above and other related works are summarized in Table 3.

5. Review of Mask-Face Detection Hardware Systems

To develop highly integrated, user-friendly and flexible mask–face detection systems, deep learning models or other software methods need to be incorporated with embedded systems such as Raspberry Pi [73]. While Wuttichai et al. [74] proved that even when
implemented in embedded systems, deep learning based approaches are more suitable than traditional machine learning methods; most of the recent research on mask-face detection systems are based on deep learning models. Due to the complexity of deep learning models, integrating mask-face detection models with hardware devices at a reasonable cost is the main problem that should be taken into consideration. However, since deep learning models require training and testing on high-end desktop hardwares or connection to remote servers with sufficient computational resources in general, most research reviewed in Section 4 are beyond testing and tend to be unable to perform in real time when deployed in embedded systems with limited acceleration devices. Based on the reasons mentioned above and insufficient existing surveys, we provide an additional review on system-level mask detection methods. In the following Section, in order to emphasize the perspective of the system-level, we review research works which either provide detailed hardware implementations or have unique contributions to mask–face detection systems. Systems that are only tested on PCs/remote servers with simple webcams or do not specify the hardware configurations are excluded.

5.1. Recent Mask-Face Detection Systems

In this section, we introduce the latest mask–face detection systems. We categorize all these research works into the following four categories: Hardware Analysis, System Packaging, Model Compression and Others according to their contributions. Definitions of these categories are as follows:

- **Hardware Analysis**: Either making a careful comparisons among various hardware types or developing special techniques to perform acceleration on the hardware platform.
- **System Packaging**: Developing ready-to-use mask-face detection product prototypes such as a smart door.
- **Model Compression**: Leveraging various compression techniques to build light-weight mask–face detection models to fit the embedded systems.
- **Others**: Contribute to mask–face detection hardware system from other perspectives.

We first begin by detailing several representative studies from each category. Then, we present a thorough summary of all the studies investigated in Table 4.
Table 4. Summary of mask-face detection hardware systems.

| Ref. & Publish Date | Hardware Platform | Speed Performance/Detection Performance | Datasets | Models | Merits | Limitations |
|---------------------|-------------------|-----------------------------------------|----------|--------|--------|-------------|
| May 2022, [75]      | a. Raspberry Pi 4 computational platform + Google Coral Edge TPU + 8-bit quantized model, b. Raspberry Pi 4 computational platform + Intel Neural Compute Stick 2 (INCS2) + Compressed model in float16 precision, c. Jetson Nano + 128 Maxwell cores + Compressed model in float32 precision | a. FPS of 19, b. FPS of 18, c. FPS of 22/ | 3843 samples of 1915 wearing masks and 1928 without masks | Made comparisons of three-well designed embedded systems in terms of both accuracy and real-time performance. Proposed a new compressed CNN model to minimize the cost. | Provided no hardware implementation details. Has yet to be integrated into a complete mask-face detection system. |
| Hardware Analysis  | July 2021, [76]   | PYNQ-Z2 all-programmable SoC platform | FPS of 45.79, Detection delay of 0.13 s/ | Medical mask dataset [77] | Two concatenated Haar Cascade Classifiers (HCC) [78] | Carefully analyzed the video delay which derives from both hardware and software. Used the multithreading technique to avoid the delay caused by the algorithm. | The mask-wearing status was judged only by the mouth detection in which the absence of mouth in the detected face was considered as mask-wearing. Such a method seems unreasonable in cases such as incorrect mask-wearing with nose exposed or obstructed faces without masks. |
| System Packaging    | March 2022, [79]  | Raspberry Pi 4 | Latency of around 90 ms (no specific value provided)/ | 1376 images of 686 wearing masks and 690 without masks | A customized CNN model | Used motor controller as gate controller and buzzer as an alarm to simulate real environment in public transportation. | There was no detailed description of the proposed model structure. Performance result was confusing (many different experimental results presented in the original paper). |
Table 4. Cont.

| Ref. & Publish Date | Hardware Platform | Speed Performance/Detection Performance | Datasets | Models | Merits | Limitations |
|---------------------|-------------------|------------------------------------------|----------|--------|--------|-------------|
| October 2020, [80]  | MiniPC embedded with GTX 1060 Nvidia GPU | FPS of 11.1/ - | - | YOLOv4 [41] | The proposed system was implemented in the real case at Batam Island. Provided photographs of application scenarios similar to the custom security check. | Lack of detection performance made it hard to evaluate the whole system quality. |
| December 2020, [81] | Personal devices (PC, tablets, mobile phones, etc.) | FPS of iPad Pro (2016 A9X): 4.12, iPhone 11(2019 A13): 8.57, MacBook Pro 15(2019 i7-9750H): 20.36/ mAP of 89% | A combination dataset of 4065 from MAFA, 3894 from WIDER FACE, 1138 from internet | YOLO-Fastest [15] | The proposed mask-face detection system was free of additional acceleration hardware, OS and device agnostics. Accessible by any user and achieves low-cost. | Detection Performance was low due to the lightness of the model and the model inference completely depended on local CPUs. |
| November 2021, [82] | Raspberry Pi tiny computer | -/ Accuracy of 99.22% | A customized dataset of 1376 images: 690 images wearing masks and 686 images without masks | A CNN-based neural network | The highly integrated system had many extended functions such as body temperature measurement and gate opening control to simulate an IoT-Enabled smart door. | The dataset was too small, and the deep learning model structure was not specified. The mask detection was based on a classification model, so it was unable to recognize multiple persons with or without masks. There was no real-time performance evaluated. |
| April 2022, [83]    | Raspberry Pi 4 Model B -8 GB Ram, Raspberry Pi camera, Raspberry Pi display, MLX90614 non-contact infrared temperature sensor, SG 90 mini servo motor | -/ Accuracy of 97.0% | 2902 images of mask-wearing and 2872 images of no mask or incorrect mask-wearing from [84] | MobileNetV2 [32] & Haar Cascade Algorithm | The whole system was of high integration. The description and visualization of its hardware system is in provided detail. Used a hybrid method which combines traditional machine learning and deep learning models to perform the mask-face detection task. | The proposed model was light-weight, but no speed performance analysis was performed to prove the value of its hybrid approach. The performance comparison presented in the paper with other mask-face detection methods was invalid because they were evaluated on different datasets. |
Table 4. Cont.

| Ref. & Publish Date | Hardware Platform | Speed Performance/ Detection Performance | Datasets | Models | Merits | Limitations |
|---------------------|-------------------|------------------------------------------|----------|--------|--------|-------------|
| November 2021, [85]  | a 5-megapixel camera, Raspberry Pi 3 model B+, buzzer and a monitor screen | FPS of 5/ Success rate of 88.89% | - | Multiple Haar Cascade Classifiers (HCC) [78] | The whole system was in low-cost. Comparing to [76], Haar Cascade Classifiers were used to detect both nose and lip so the proposed method can detect incorrect mask-wearing faces. | The speed performance was too low to perform mask-face detection in real-time. There was no dataset source provided; therefore, the accuracy was not convincing. |
| October 2020, [86]   | Coral Dev Board embedded with a Google Edge TPU coprocessor | Latency of 6.4 ms/ Accuracy of 58.4% | A customized dataset of 2127 images | MobileNetV2 [32] + SSD | Showed the complete progress of shrinking the model size. | The accuracy lost about 6% points while comparing the 8-bit integer model to the original 32-bit model according to the experiment of 1NN model on CPU in the original paper. |
| September 2020, [87] | Pynq Z1 platform (with an ARM processor and a single-chip FPGA) | FPS of 16/ Precision of 90.7%, Recall of 92.3% | A Combination of Real-World Masked Face Dataset (RMFD), Masked Face Detection Dataset (MFDD), Simulated Masked Face Recognition Dataset (SMFRD), Masked Faces (MAFA) dataset | Pynq-YOLO-Net (a pruned and quantized YOLO model) | Lower decrease in performance compared to other state-of-the-art models while applying a dedicated compression strategy to the model. | Lacked a comparison of acceleration performance between before compression and after compression. The effectiveness of the proposed compression strategy was not illustrated. |
| May 2021, [88]       | A small development board with an ARM Cortex-M7 microcontroller clocked at 480 Mhz and 496 KB of framebuffer RAM | FPS of 30/ Accuracy of 99.83% | 14,211 images in wearing mask and without mask from Kaggle and openCV | A customized CNN model | The proposed model was super light-weighted of 138 KB after integer quantization. Outperformed Squeeze Net which was a similar size. | The model only acted as a two-class classifier, not a detector. It could not be developed to perform two-class detection tasks due to the choice of BinaryCrossentropy loss. |
Table 4. Cont.

| Ref. & Publish Date | Hardware Platform | Speed Performance/ Detection Performance | Datasets | Models | Merits | Limitations |
|---------------------|-------------------|------------------------------------------|----------|--------|--------|-------------|
| June 2021, [89]     | A low-power, real-time embedded FPGA accelerator based on the Xilinx FINN architecture | FPS of 6460, Latency of 0.31 ms/ Accuracy of 93.94% | MaskedFace-Net dataset with a detailed split into 4 classes: Correctly Masked Face Dataset (CMFD), Incorrectly Masked Face Dataset (IMFD) Nose, IMFD Chin, and IMFD Nose and Mouth | BinaryCop—a binary neural network (BNN) classifier | The system was capable of executing a finer classification. The inference was fast with the combination of a Binary Neural Network (BNN) and an acceleration framework. The embedded system hardware required low power. | Remains to be tested in other platforms except embedded FPGAs. |
| June 2022, [90]     | Raspberry Pi 4    | FPS of 1.97/ mAP of 57.40%               | Mask-Detection-Dataset [91] | YOLOv4-tiny [41] | Provided an analysis on multiple strategies of model pruning and quantization. Achieved a great improvement in FPS while a decrease in mAP was observed. | The pruning and quantization analysis was not theoretical; thus, it was hard to say that the best strategy for YOLOv4-tiny could be generalized to other deep learning models. The FPS was still so low even after compression, showing that the system is not yet suited to real-use implementation. |
| December 2021, [92] | a. Nvidia Jetson Nano, b. Jetson Xavier NX | a. FPS of 12 and 8 for YOLOv5s and YOLOv5l; b. FPS of 30 and 24 for YOLOv5s and YOLOv5l/ mAP of 86.43 and 92.49 for YOLOv5s and YOLOv5l | A customized dataset of 3000 images divided into 3 classes (mask-wearing, no mask and incorrect mask-wearing) | YOLOv5s, YOLOv5l | The proposed system was capable of detecting a third class of Improper Mask-wearing. | There was no contribution of fitting deep learning models to the hardware system, leading to a high cost of hardware. |
| Ref. & Publish Date | Hardware Platform | Speed Performance/Detection Performance | Datasets | Models | Merits | Limitations |
|-------------------|-------------------|-----------------------------------------|----------|--------|--------|-------------|
| October 2020, [74] | Raspberry Pi4 with intel Neural Computer Stick 2 | -/ Accuracy of 88.7% | Prajna Bhandary dataset of 1376 images: 690 images wearing masks and 686 images without masks | MobileNet [29] | Presented a comparison among deep learning models (MobileNet) and traditional machine learning methods (SVM and KNN). | Model was trained with a relatively small dataset. Hardware performance was not tested. |
| March 2021, [93]  | Raspberry Pi      | System latency time of 0.13 s/ mAP of 73.7% | Properly Wearing Masked Face Detection Dataset (PWMFD) including 9205 images with 3 categories (mask-wearing, no mask and incorrect mask-wearing) | Squeeze and Excitation (SE)-YOLOv3 | Established a complete detection system as an access control gate. Able to detect a third class of incorrect mask. | The proposed model SE-YOLOv3 had more parameters than YOLOv3 which negatively affected the inference performance. Further experiments on other types of hardware device and model simplification remained to be conducted. |
5.1.1. Hardware Analysis

Peter et al. [75] provided an elaborate comparison among multiple hardware choices. They set a desktop-based implementation of an Intel i7-8650U CPU 1.9 GHz, and 16 GB RAM as the baseline. Then three embedded hardware systems were designed to deploy the deep learning detection model. From the software level, the authors proposed a customized CNN model named MaskDetect. Compared to well-known pretrained models such as VGG-16 [44] and ResNet-50V2 [71], MaskDetect was light-weight, thus saving memory and accelerating the inference to a great extent. Furthermore, to fit the specific hardware accelerator, the author provided three different quantization strategies. A memory reduction from the model size of 11.5 MB to 0.983 MB can be realized under the 32-bit floats to 8-bit integer quantization compression. As a result, three systems, namely the a. Raspberry Pi 4 computational platform + Google Coral Edge TPU + 8-bit quantized model, b. Raspberry Pi 4 computational platform + Intel Neural Compute Stick 2 (INCS2) + Compressed model in float16 precision and c. Jetson Nano + 128 Maxwell cores + Compressed model in float32 precision achieved the following performances: FPS—19, accuracy—90.4%; FPS—18, accuracy—94.3% and FPS—22, accuracy—94.2%, respectively. The NVIDIA Jetson Nano is least costly in terms of design and had the best performance of acceleration while its detection accuracy was comparable. This paper is the only research in which multiple systems were designed and an in-depth comparison was made from different angles.

In [94], the authors provided a similar result to [75] that the NVIDIA Jetson Nano might be a more suitable platform for developing mask-face detection systems. They deployed various deep learning models of both image classification and object detection on Jetson Nano and Raspberry Pi3 for mask-face detection and performed a real-time analysis. Consequently, the Jetson Nano outperformed Raspberry Pi3 by a substantial degree. As an example, Jetson Nano provided 36 FPS with the well-known ResNet50 [28] while the FPS was 1.4 for Raspberry Pi3 in the same configuration. As a result, though most systems tend to deploy models on Raspberry Pi embedded systems, the above research revealed that Raspberry Pi might not be the most suitable choice for designing mask-face detection systems. More hardware devices should be explored.

Besides looking for suitable hardware devices, Fang et al. [76] achieved great progress in addition to designing detection models. In their approach, they used the multithreading technique to avoid the model delay. They trained Haar Cascade Classifiers (HCC) [78] to detect mask-wearing and faces without a mask. They used two HCCs to detect faces and months, respectively. The mask-face detection results were based on a combination of two outcomes. While no deep learning models were deployed, it was proved that the traditional machine learning model could cause significant delay due to the algorithm’s running time. To avoid such delays, they incorporated multithreading whereby they pipelined the video frame through the FPGA unit and executed the detection model in the ARM core. Therefore, the algorithm delay could be replaced by a buffer delay which improved the FPS to a large extent.

5.1.2. System Packaging

Susanto et al. [80] put their mask-face detection system into use at Politeknik Negeri Batam island. They installed a MiniPC with GPX 1060 Nvidia GPU together with a digital webcam and screen to assemble the system. The authors of [82,83] integrated their systems into smart and automatic doors. The doors were driven by servo motors and determined whether a person could enter or not by performing mask-face detection and checking body temperature.

Unlike the previously mentioned approaches based on embedded systems, Wang et al. [81] proposed an in-browser serverless edge-computing based mask-face detection system in which all computations were executed by personal devices. It is so flexible that users can access their server through their mobile phones or PC at anytime and from anywhere. Another advantage is that the abstract system is of little cost since it is device-agnostic meaning it requires no additional hardware platform. However, be-
cause the detection model inference is operated by CPUs on personal devices, the real-time performance is not satisfying. Moreover, whether such an online system could be applied to real cases remains to be discovered.

5.1.3. Model Compression

Aiming at compatibility with embedding systems and real-time implementation, various methods were proposed to compress the deep learning model. The studies in [75,86–88,90] all applied quantization to the deep learning models in order to fit the model into embedded systems and achieve faster inference performance. Among them, Ref. [90] provides several ablation studies on both pruning and quantization procedures. It provides a guideline for object detection model compression with the strategy of a 0.6 pruning rate and static quantization. In addition, Yahia Said [87] raised that since mask-face detection is a binary classification problem (i.e., wearing mask and without mask), the process of class probability calculation could be eliminated to lessen the inference time. They treated the output confidence scores as class probabilities to avoid additional forward calculation.

Nael et al. [89] presented a special quantization method named BinaryCop to reduce the model memory footprint. The proposed model was based on the Binary Neural Network (BNN) founded by [95]. BNN can be considered as an extreme case of quantization. It quantizes all parameters of weights and activations to one-bit precision as $-1$ and 1. Consequently, most complicated calculations in the neural network can be replaced by binary operations such as XOR during the forward pass of the neural network. As such, a unique structure could be accelerated by performing deployment on a delicately designed hardware; the authors implemented their BinaryCop models on embedded FPGAs based on the Xilinx FINN framework [96], which was designed for FPGA accelerations of BNNs. When pre-processing on the dataset, they subdivided the MaskedFace-Net dataset’s [84] two classes of Correctly Masked Face Dataset (CMFD) and Incorrectly Masked Face Dataset (IMFD) into four classes of CMFD, IMFD Nose, IMFD Chin, and IMFD Nose and Mouth. From the experiments, three proposed BinaryCop models of CNV, n-CNV and µ-CNV achieved a latency of 1.58 ms, 0.31 ms, 0.81 ms and accuracy of 98.10%, 93.94%, 93.78%, respectively. The comparable results proved the effectiveness of BinaryCop and its requirement of a low inference power of 2 W.

Besides employing quantization to generate light-weight models, [83] proposed a hybrid approach to perform the mask-face detection task. The architecture was divided into two stages. The first stage exploited the Haar Cascade algorithm [78], also called the Voila–Jones algorithm to detect faces. It is a traditional machine learning algorithm used to carry out object detection and therefore requires fewer computational resources. The authors used pre-trained face detection Haar Cascade models in OpenCV in their method. In the second stage, they trained a MobileNetV2 [32] deep learning model from scratch as the classifier of two classes (mask-wearing and no mask/incorrect mask-wearing). From their experiments, the hybrid model free of a deep learning based detector could achieve a comparable performance.

5.1.4. Others

Other research works contribute to mask-face detection system design from different angles. For instance, Ref. [74] studied the performance of K-nearest neighbors (KNN), support vector machine (SVM) and MobileNet [29] when performing mask-face detection tasks on Raspberry Pi 4. They showed that even in a real-time case, a light-weight version of the MobileNet algorithm can achieve a better accuracy of 88.7% than both the SVM and KNN algorithms (78.1% and 72.5%, respectively).

5.2. Discussions

Regarding the systems summarized above, we provide specific evaluations of both hardware acceleration performance and software detection performance for each study in
Table 4. According to the models in Table 4, we can easily see that all approaches using traditional machine learning methods are based on Haar Cascade Classifiers (HCC) since HCC is a popular machine learning algorithm to perform object, especially face, detection. Nevertheless, according to [97], based on the fact that the HCC has weaknesses in lowlight conditions, pure HCC requires an additional IR (Infrared) module on the camera to improve its performance. Meanwhile, models based on deep learning are able to outperform HCC-based methods without any additional hardware device. Therefore, these results further prove the conclusion that deep learning-based masked-face detection methods are mainstream.

As illustrated in Figure 3a, most research works focus either on packaging the systems into ready-to-use products or compressing masked-face detection models to perform in real-time on the hardware platform. Only a few papers contribute to the hardware design to achieve a better speed performance or make a comparison among different devices in detail. Such a conclusion provides potential in terms of the masked-face detection system for future studies. In addition, we visualized the proportions of two-class and three-class mask-face detection systems in Figure 3b. Particularly, we categorized the four-class classification model [89] into a three-class model since two classes of incorrectly masked nose and incorrectly masked chin can both be regarded as incorrect mask-wearing. As shown in the figure, three-class mask-face detection is performed less frequently on hardware systems. This is because preparing a three-class mask-face detection model is of higher-cost and more time-consuming. For example, labelling the third class of incorrect mask-wearing in the dataset is costly and may easily cause a data imbalance. Under such circumstances, in the next section, we introduce an approach to quickly extract abnormal cases of incorrect mask-wearing which can be incorporated into all two-class mask-face detection systems with no additional data required.

6. OOD-Mask Detection Framework

In this section, we propose a new Out-of-distribution Mask (OOD-Mask) framework to detect the abnormal class of incorrect mask-wearing with only two-class data. Due to its simplicity and decoupling, large numbers of existing two-class mask-face detection models can be directly incorporated with the OOD framework to perform three-class mask-face detection tasks.

6.1. Related Works

6.1.1. Out-of-Distribution Detection

In supervised deep learning models, the number of classes that models are able to classify is consistent with the number of training labels. However, a special technique such as the Out-of-Distribution (OOD) detection method can detect an additional class not included
in the training dataset. It is usually closely applied in anomaly detection (AD) [98], novelty detection (ND) [99], open set recognition (OSR) [100], and outlier detection (OD) [101].

Several advanced methods were presented to ensure OOD abnormal detection performance while the original model performance was not affected. Hendrycks et al. The study in [102] proposed an approach to effectively detect OOD samples by predicting the softmax probability of the sample through a statistical model. Liang et al. [103] proposed the use of temperature scaling and input processing to improve the OOD detection capability. Denouden et al. [104] detected OOD samples by exploiting the reconstruction error of the variational autoencoder (VAE) [105]. Abdelzad et al. [106] extracted the input and output data of different layers, used a one-class SVM classifier, counted the classification error rate of this layer, and then selected the layer with the smallest error to detect OOD samples.

Based on the characteristics of the OOD detection approach, we found it suitable for the mask-face detection task where the data of incorrect mask-wearing are hard to collect and annotate. We were then motivated to develop a new framework named Out-of-Distribution Mask (OOD-Mask) incorporating the state-of-the-art object detection neural network, which is aimed at carrying out the three-class mask-face detection task.

6.1.2. YOLOX Object Detection Model

The YOLO series is one of the most widely used detectors in the industry due to its broad compatibility, so we selected the YOLOX [107] architecture, which is a high-performance anchor-free detection model, as the backbone detector in our framework.

Despite the fact that the original YOLOX model is one of the state-of-the-art object detection models, it shares the common limitation of supervised neural networks that the numbers of training classes and prediction classes have to be the same so that classes not included in training datasets are not detected.

YOLOX uses YOLOv3 as the baseline, which adopts the architecture of the DarkNet53 backbone and an SPP (Spatial Pyramid Pooling) space [108]. The SPP structure performs feature extraction through the maximum pooling of different pooling kernels to improve the perception field of the network. After that, the three enhanced feature layers are passed into Yolo’s head to obtain the prediction results. The classification and regression are implemented separately, and they are integrated with the final prediction, as shown in Figure 4. The network can only classify the categories that have been labeled in the training dataset.

![YOLOX architecture](image)

**Figure 4.** YOLOX architecture.

6.2. Methodology of OOD-Mask

We proposed the OOD-Mask framework using Out-of-Distribution detection for detecting incorrect mask-wearing, according to the differences in confidence scores between in-and Out-of-Distribution. Such difference enables OOD-Mask to separate the third class of incorrect mask-wearing from the output prediction of neural networks that only contain input data of wearing mask and without mask. In the following sections, we first describe how to use the sigmoid score to detect the third class, then introduce the energy score and use it to implement the OOD-Mask for detecting incorrect mask-wearing.
6.2.1. Sigmoid Score

In order to achieve the goal of detecting the third class of incorrect mask-wearing from the samples predicted as wearing mask or without mask, we first treated both classes of wearing mask and without mask within the dataset as in-distribution (ID) while the class of incorrect mask-wearing was treated as out-of-distribution (OOD). Then, we tried to separate OOD data from ID data based on the sigmoid confidence score.

The sigmoid function is one of the most commonly used non-linear activation functions in neural networks. All real numbers are within the domain of sigmoid functions, and the return value is mapped in the range of 0 to 1. The formula is as follows:

\[ S(x) = \frac{1}{1 + e^{-x}} \]  

(9)

where the input \( x \) is a scalar.

We can obtain sigmoid scores from the decoupled head with the following three branches in YOLOX: class, object, and regression corresponding to the prediction scores of the target box class, the score of whether the box has the target, and the predicted bounding box in order. Multiplication is performed to obtain the final sigmoid score after sending class and object through the sigmoid function, respectively. Figure 5a shows the sigmoid score distribution of the three classes of wearing mask, without mask, and incorrect mask-wearing. As illustrated in the figure, the peak value of incorrect mask-wearing (OOD) overlaps with that of wearing mask and without mask (ID), and the OOD data is uniformly distributed in the overall dataset. In addition, because the output value activated by the sigmoid function ranged between 0 and 1 and the data during the experiment were taken to three decimal places, it was difficult to make more accurate selections. Therefore, extracting the OOD data while remaining the performance of ID remains unchanged seems impossible under such circumstance.

![Sigmoid score](image)

**Figure 5.** Data distribution: (a). The peak of the incorrect mask-wearing which is at around \( S = 0.8 \), almost overlaps the peaks of both wearing mask and without mask in ID, although the class has a large proportion of values in the 0.2–0.7 range. (b). the values obtained by the OOD-Mask are all negative numbers. In order to compare with the sigmoid score, we take the absolute value of the energy score. the spiky distribution of the incorrect mask-wearing, at absolute energy score \( |E| = 19.3 \), is completely separated from wearing mask and without mask, at \( |E| = 17.3 \).

6.2.2. Energy Score

To separate the distribution of OOD data from the ID, we introduced the complete OOD-Mask framework as illustrated in Figure 5b. OOD-Mask can attribute lower values to in-distribution data and higher values to out-of-distribution data. During the testing process, the detector of YOLOX with trained weights and parameters are combined with OOD-Mask to perform the detection task. We input the whole testing set into the model, which includes three classes (wearing mask, without mask and incorrect mask-
wearing). The backbone detector is responsible for predicting two classes (wearing mask, without mask) because it is trained by the training dataset of only two classes. Then, its prediction is passed to the OOD-Mask to extract the third class of incorrect mask-wearing. Within the OOD-Mask, we first use class and object branches from the decoupled head of YOLOX to calculate both the sigmoid score and energy score separately. The energy score is implemented to classify the ID classes and the OOD class. If the sample belongs to the ID classes, it can be further classified by the sigmoid score. We detailed the method of the energy score as follows.

The energy function \[109\], first proposed by Weitang Liu et al. plays an important role in this method, which builds upon earlier work \[110\] relating to the energy-based Model (EBM). EBM has an inherent connection with modern machine learning, especially discriminate models. The energy function creates a discriminative neural classifier \(f(x)\): \(\mathbb{R}^K \rightarrow \mathbb{R}\), which maps the \(K\) classes of input space to a single, non-probabilistic scalar called the energy. The formula of the energy function is as follows:

\[
E(x; f) = -T \times \log \sum_{i=1}^{K} e^{f_i(x)/T}
\]  

(10)

where \(T\) is the temperature parameter, and the subscript \(i\) indicates the \(i\)th class of \(f(x)\).

The energy score is used to sort the overall identified targets and extract a portion of them as OOD data. With two types of scores, we not only maintain the accuracy of the wearing mask and without mask classification in the ID but also provide more meaningful information to maximize the distinction between in-distribution and out-of-distribution. The two-class detection prediction of the backbone detector is passed through the OOD-Mask module and the three-class detection result can be obtained. The energy score leads to an overall smoother distribution and can largely separate the class of incorrect mask-wearing from other labels without reducing the performance of other ID classes. The complete OOD-Mask detection framework and its inference process are shown in Figure 6, where the YOLOX detector is trained using the two-class dataset.

![Figure 6](image.png)

**Figure 6.** Network Architecture: The YOLOX network is trained with two-class training dataset and performs two-class detection. The prediction output of YOLOX is then passed through the OOD-Mask to obtain three-class detection results.

Additionally, compared to only using the sigmoid function for classification, the energy function does not perform normalization and directly maps each input to a single scalar, which results in lower energy values for in-distribution and higher energy for incorrect mask-wearing, making the gap between categories more obvious.

### 6.3. Dataset

Our mask detection and classification dataset was divided into three categories: (1) wearing mask, (2) without mask, and (3) incorrect mask-wearing. The sample source in this paper was taken from Wider Face [16] and Kaggle.

Our proposed dataset contains a total of 1174 images and annotations in a Pascal VOC format. The training dataset contains 3000 wearing mask, and 3322 without mask,
and there are 223 wearing mask, 268 without mask and 157 incorrect mask-wearing in the testing set. The above numbers illustrate that the number of people wearing masks incorrectly in the dataset is limited and much lower than the other two classes. Figure 7 shows several examples included in the dataset.

Figure 7. Examples within the dataset.

6.4. Experiments & Results

We used YOLOX as the object detection model. The training process was carried out on a server with an NVidia RTX 2080 Ti video card. The training configuration was set in a batch size of 16 and 200 epochs, and the entire training process lasted 52.38 min. To highlight the effectiveness of the proposed OOD-Mask framework, we carried out three experiments for comparison as follows.

6.4.1. Experiment on the Original Model

The experiment on the original model without OOD-Mask for two-class mask-face detection acts as the baseline. We used criteria of recall, precision, and F1-score to evaluate the two-class detection performance when the value of the intersection of union (IoU) was 0.5, which had the best classification effect. Results are shown in the first row of Table 5.

Table 5. Experiment Results.

| Train | Test | OOD-Mask | Wearing_MASK (%) | Without_MASK (%) | Incorrect_MASK-Wearing (%) |
|-------|------|----------|------------------|------------------|---------------------------|
|       |      |          | Precision | Recall | F1 | Precision | Recall | F1 | Precision | Recall | F1 |
| 1     | 2-cls| 2-cls    | X        | 74.69   | 82.06 | 78.21 | 72.20   | 84.33 | 77.80 | - |
| 2     | 2-cls| 3-cls    | X        | 81.41   | 72.65 | 76.96 | 78.65   | 78.36 | 78.50 | 40.40 | 25.48 | 31.25 |
| 3     | 2-cls| 3-cls    | ✓        | 83.08   | 72.65 | 77.51 | 82.04   | 75.00 | 78.36 | 53.18 | 53.60 | 55.76 |

2-cls: data of wearing mask and without mask; 3-cls: data of wearing mask, without mask and incorrect mask-wearing.

6.4.2. Experiment on Sigmoid Scores

In the second experiment, in order to extract the OOD data within the class of incorrect mask-wearing, the sigmoid confidence scores in the original network were used to correspond to the different distributions of the three classes (wearing mask, without mask, and incorrect mask-wearing). We used the test dataset of three-class to evaluate the performance while keeping the performance of ID data as constant as possible. It was shown that the data peaks of the three classes were all stacked around a sigmoid score of 0.8, but there was still a large part of the OOD data scattered in the range of less than 0.7. If 0.7 was used as the threshold, value if less than 0.7 could be regarded as OOD, and values greater than this were regarded as ID. However, this demarcation method reduced the performance of wearing mask and without mask greatly. After changing the score range many times, we chose the data with a sigmoid score of between 0.5 and 0.67 as the OOD by observing three sigmoid score distributions. The evaluation results of the second experiment are
presented in Table 5. While the performance of wearing-mask and without-mask remained unchanged, the best F1-score for incorrect mask-wearing was only 31.25%, which meant the extraction was not satisfactory.

6.4.3. Experiment on OOD-Mask Detection Framework

The third experiment was based on the framework combining the YOLOX detection head and OOD-Mask, which used the energy function to score the results without reducing the performance of wearing mask and without mask. It separated the OOD data from the ID to a greater extent, so the distinction effect of the OOD could be improved. We used both energy score and sigmoid score in OOD-Mask and changed the range of the energy score by adjusting the value of T in the energy function, as shown in Figure 8. A comparable classification performance could be achieved at a low cost. When T equaled 30, the energy score range was between 14 and 22. When we observed the energy score distributions of three classes, setting the threshold to 18.3 provided a greater degree of discrimination between abnormal and normal classes. Therefore, after many experiments, we selected an energy score of between 18.3 and 22 as the OOD, and the rest were divided into the ID. The wearing mask and without mask classes in ID were still classified using the sigmoid function.

![Energy score distributions for different T values](image)

**Figure 8.** Class distributions corresponding to different T values: (a) When T equals 1, the overlap of three distributions makes it challenging to extract the incorrect mask-wearing class. (b) When T is set to 10, the peak of incorrect mask-wearing distribution can be clearly distinguished from other classes (wearing mask and without mask). (c) When T is set to 30, the distinction effect is further improved. (d) When the value of T is greater than or equal to 30, the distribution of the three classes changes little. The energy scores increase with the value of T, but the range of score still remains within 10.
It was shown that the results of this method’s F1-score improved by 24.51% over the results of the second experiment. Compared with the previous experiment, the values of precision and recall improved by 13% and 28%, respectively, as shown in the third row of Table 5. Besides, the performance of both wearing mask and without mask was comparable. Under the same condition as in the second experiment that only two classes (wearing mask and without mask) of data were input into the training process, without any data support for the third class of incorrect mask-wearing, the improvement in the detection performance of the third class was considerable. However, since there was no training data for the third class, the performance still had room for further improvement. Therefore, new OOD-related variants remain to be discovered.

Furthermore, to assess the performance of experiments 2 and 3 more thoroughly, we classified wearing mask and without mask as in-distribution (positive class) and incorrect mask-wearing as out-of-distribution (negative class). The receiver operating characteristic (ROC) curve is a performance measure for classification problems under various threshold settings, where the y-axis is TPR and the x-axis is FPR, and area under curve (AUC) is the area under the ROC curve, indicating the degree or measure of separability. Since some targets were not recognized by the network due to being too small or inconspicuous features, the TPR and FPR values could not reach 1.0. For the AUC value, 53.43% in experiment 3 was higher than 45.80% in experiment 2. By observing the ROC of the two experiments in Figure 9, it was shown that the classification performance of using the energy score was more excellent and obvious than only using the sigmoid score. When we took the same FPR value of between 0.2 and 1.0, the value of TPR in the energy score was consistently higher than in the sigmoid score.

![Figure 9. ROC under different scores.](image)

In order to test the speed performance, we divided all images in the testing set by the whole processing time and used the frame per second (FPS) as the evaluation metric. The test dataset contained a total of 235 images. As shown in Table 6, the frame rate of the three experiments reached 76.18, 65.68 and 64.51 FPS, respectively. The model with OOD-Mask resulted in slightly lower performance due to its additional calculation on data distribution. However, the calculation complexity was limited, and the performance loss was acceptable since the OOD-Mask still met the real-time detection requirement.
Table 6. Speed Performance.

| Experiment | GPU  | FPS  |
|------------|------|------|
| 1          | 2080Ti | 76.18 |
| 2          | 2080Ti | 65.68 |
| 3          | 2080Ti | 64.51 |

The experiment numbers correspond to the following three experimental chapter numbers, respectively.

7. Conclusions

In this work, we first presented an up-to-date review on two-class and three-class mask-face detection models separately to extract the common characteristics and main distinction between the two mentioned fields. We discussed the state-of-the-art approaches and found that researchers prefer one-stage object detection techniques in both two-class and three-class fields for the real-time requirement of masked-face detection applications. We also deduced that research works on three-class masked-face detection models provided new insights, mainly in terms of extending the dataset with more training samples of incorrect mask-wearing or improving models, such as introducing the attention mechanism.

A survey on mask-face detection hardware systems was then provided. We summarized that recent hardware system research and mostly contributed in the following three aspects: hardware analysis, system packaging, and model compression. Notably, emphasis of research in this part varied, and no study paid attention to multiple aspects. Therefore, a system which both builds a light-weighted and well-performing detection model and designs a well-adapted and user-friendly hardware platform remains to be developed.

Finally, we proposed a new framework named Out-of-Distribution Mask (OOD-Mask). The framework aimed to extract the abnormal cases (i.e., incorrect mask-wearing) from wearing mask and without mask. Thus, the proposed framework succeeded in performing the three-class mask-face detection task based on the data and model required in the two-class mask-face detection task.

Developing an integrated mask-face detection system that is low cost, has a fast response, and high accuracy requires techniques in various aspects as we discussed. Based on the survey we conducted, we suggest some research directions for future work. In order to ensure that the detection system can identify whether a person is wearing a mask correctly, researchers need to pay more attention to three-class mask-face detection models. Moreover, in order to make these models compatible with the corresponding hardware systems, instead of increasing model complexity for better accuracy, detection performance should be improved on light-weight models. The OOD framework presented in our work is capable of achieving this, but performance needs to be improved in future work. Furthermore, since the difference between wearing a mask correctly and incorrectly is rather subtle and may be formulated as a fuzzy problem, new fuzzy models [111] could be adopted for this problem in the future. On the hardware side, more hardware research and comparisons are required on how to choose a suitable hardware device to deploy a light-weighted deep learning model. In addition, improving compatibility and achieving better hardware acceleration performance are also significant research directions. Overall, we expect a combination of state-of-the-art techniques both in three-class mask-face detection models and hardware design, as well as continuous efforts in each domain.

Author Contributions: Conceptualization, Y.H., Y.X. and H.Z.; methodology, Y.H. and Y.X.; software: Y.H.; validation: Y.X.; formal analysis, Y.H. and Y.X.; investigation, Y.H. and Y.X.; resources, Z.W., Z.L. and H.Z.; data curation, Y.H. and H.Z.; writing—original draft preparation, Y.H., Y.X. and Z.L.; writing—review and editing, Y.X., Y.H., Z.L., H.Z. and Z.W.; visualization, Y.X.; supervision, Z.L., H.Z.; project administration, H.Z. and Z.W.; funding acquisition, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Engineering Research Council, Agency of Science, Technology and Research, Singapore, through the National Robotics Program under Grant No. 1922500054.
**Institutional Review Board Statement:** Ethical review and approval were waived for this study due to the images/photos used in this paper deriving from the internet which is openly available.

**Informed Consent Statement:** Not available.

**Data Availability Statement:** The photos in this study were taken from the publicly archived dataset, Face Mask Detection, [https://www.kaggle.com/datasets/andrewmvd/face-mask-detection](https://www.kaggle.com/datasets/andrewmvd/face-mask-detection) (last accessed in June 2022); Wider Face, [http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/](http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/) (last accessed in June 2022), and the publicly available link: [http://www.latender.com/Sleep_health/1159.html](http://www.latender.com/Sleep_health/1159.html) (last accessed in July 2022).

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

**Abbreviations**

The following abbreviations were used in this manuscript:

| Abbreviation | Description                                      |
|--------------|--------------------------------------------------|
| COVID-19     | Coronavirus disease-2019                         |
| WHO          | World Health Organization                        |
| PHEIC        | Public Health Emergency of International Concern |
| TP           | True Positive                                    |
| TN           | True Negative                                    |
| FP           | False Positive                                   |
| FN           | False Negative                                   |
| TPR          | True Positive Rate                               |
| FPR          | False Positive Rate                              |
| IoU          | Intersection over Union                          |
| AP           | Average Precision                                |
| mAP          | mean Average Precision                           |
| HOG          | Histogram of Oriented Gradient                    |
| LBP          | Local Binary Pattern                             |
| CNN          | Convolutional Neural Network                     |
| GAN          | Generative Adversarial Network                   |
| ResNet       | Residual Neural Network                          |
| YOLO         | You Only Look Once                               |
| SSD          | Single Shot Detector                             |
| RPN          | Region Proposal Network                          |
| R-CNN        | Region-based Convolutional Neural Network        |
| CBAM         | Convolutional Block Attention Module             |
| MAFA         | Masked Face                                      |
| MFDD         | Masked Face Detection Dataset                    |
| SMFRD        | Simulated Masked Face Recognition Dataset        |
| RMFRD        | Real-world Masked Face Recognition Dataset       |
| RMFD         | Real-World Masked Face Dataset                   |
| FMD          | Facemask Detection                               |
| MAFA-FMD     | MAsked FAces for Face Mask Detection             |
| WMD          | Wearing Mask Detection                           |
| FM           | Face Mask                                        |
| FPS          | Frame Per Second                                 |
| BNN          | Binary Neural Network                            |
| FPGA         | Field Programmable Gate Array                    |
| HCC          | Haar Cascade Classifier                          |
| SVM          | Support Vector Machine                           |
| KNN          | K-NearestNeighbor                                |
| AD           | Anomaly Detection                                |
| ND           | Novelty Detection                                |
| OSR          | Open Set Recognition                             |
References

1. WHO Coronavirus (COVID-19) Dashboard. 2022. Available online: https://covid19.who.int/ (accessed on 1 July 2022).
2. Cao, Y.; Yisimayi, A.; Jian, F.; Song, W.; Xiao, T.; Wang, L.; Du, S.; Wang, J.; Li, Q.; Chen, X.; et al. BA. 2.12. 1, BA. 4 and BA. 5 escape antibodies elicited by Omicron infection. *Nature* 2022, 608, 593–602. [CrossRef]
3. Mandatory Face Masks Reintroduced at Manx Care Sites after COVID Spike. 2022. Available online: https://www.bbc.com/ news/world-europe-isle-of-man-61873373 (accessed on 1 July 2022).
4. Nowrin, A.; Afroz, S.; Rahman, M.S.; Mahmud, I.; Cho, Y.Z. Comprehensive review on facemask detection techniques in the context of COVID-19. *IEEE Access* 2021, 9, 106839–106864. [CrossRef]
5. Mbunge, E.; Simelane, S.; Fashoto, S.G.; Akinnuwesi, B.; Metfula, A.S. Application of deep learning and machine learning models to detect COVID-19 face masks-A review. *Sustain. Oper. Comput.* 2021, 2, 235–245. [CrossRef]
6. Vibhuti; Jindal, N.; Singh, H.; Rana, P. Face mask detection in COVID-19: A strategic review. *Multimed. Tools Appl.* 2022. [CrossRef] [PubMed]
7. Mohammed Ali, F.A.; Al-Tamimi, M.S. Face mask detection methods and techniques: A review. *Int. J. Nonlinear Anal. Appl.* 2022, 13, 3811–3823.
8. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* 2004, 57, 137–154. [CrossRef]
9. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
10. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
11. Chollet, F. Xception: Deep Learning With Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
13. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. 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.
17. 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.
18. 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.
19. 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]
20. Chiang, D. AIZOOTech. 2021. Available online: https://github.com/AIZOOTech/FaceMaskDetection (accessed on 1 July 2022).
21. Face Mask Detection Dataset. Available online: https://www.kaggle.com/omkargurav/face-mask-dataset (accessed on 12 December 2021).
53. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *IEEE Trans. Cybern.* **2022**, *52*, 8574–8586. [CrossRef]

54. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cham: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 8–14 September 2018; pp. 3–19.

55. Oumina, A.; El Makfhî, 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 (ICECOCOS)*, Kenitra, Morocco, 2–3 December 2020; pp. 1–5. [CrossRef]

56. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.

57. 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]

58. Han, Z.; Huang, H.; Fan, Q.; Li, Y.; Li, Y.; Chen, X. SMD-YOLO: An efficient and lightweight detection method for mask wearing status during the COVID-19 pandemic. *Comput. Methods Programs Biomed.* **2022**, *221*, 108888. [CrossRef] [PubMed]

59. Guo, S.; Li, L.; Guo, T.; Cao, Y.; Li, Y. Research on Mask-Wearing Detection Algorithm Based on Improved YOLOv5. *Sensors* **2022**, *22*, 4933. [CrossRef] [PubMed]

60. 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]

61. Kayali, D.; Dimillier, K.; Sekeroglu, B. Face Mask Detection and Classification for COVID-19 using Deep Learning. In *Proceedings of the 2021 International Conference on INnovations in Intelligent Systems and Applications (INISTA)*, Kocaeli, Turkey, 25–27 August 2021; pp. 1–6. [CrossRef]

62. Yu, J.; Zhang, W. Face Mask Wearing Detection Algorithm Based on Improved YOLO-v4. *Sensors* **2021**, *21*, 3263. [CrossRef]

63. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 7–9 July 2015; Proceedings of Machine Learning Research; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 2015; Volume 37, pp. 448–456.

64. Razavi, M.; Alikhani, H.; Janfaza, V.; Sadeghi, B.; Alikhani, E. An automatic system to monitor the physical distance and face mask wearing of construction workers in COVID-19 pandemic. *SN Comput. Sci.* **2022**, *3*, 27. [CrossRef]

65. Kumar, A.; Kalia, A.; Verma, K.; Sharma, A.; Kaushal, M. Scaling up face masks detection with YOLO on a novel dataset. *Optik* **2021**, *239*, 166744. [CrossRef]

66. Mokeddem, M.L.; Belahcene, M.; Bourennane, S. Yolov4FaceMask: COVID-19 Mask Detector. In *Proceedings of the 2021 1st International Conference On Cyber Management And Engineering (CyMaEn)*, Hammamet, Tunisia, 26–28 May 2021; pp. 1–6. [CrossRef]

67. Tomás, J.; ReGo, A.; Viciano-Tudela, S.; Lloret, J. Incorrect facemask-wearing detection using convolutional neural networks with transfer learning. *Healthcare* **2021**, *9*, 1050. [CrossRef]

68. Kumar, A.; Kalia, A.; Sharma, A.; Kaushal, M. A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *14*, 1–14. [CrossRef]

69. Eyikouk, F.I.; Ekenel, H.K.; Waibel, A. Unconstrained Face-Mask & Face-Hand Datasets: Building a Computer Vision System to

70. van Roy, B.; Zemel, R.; Alemi, A.A. Improving the robustness of deep networks using generalization-guided regularization. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 7–9 July 2015; Proceedings of Machine Learning Research; Bach, F., Blei, D., Eds.; PMLR: Lille, France, 2015; Volume 37, pp. 448–456.

71. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In *Proceedings of the Computer Vision—ECCV*, Munich, Germany, 8–14 September 2018; pp. 3–19.

72. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 15–20 June 2019.

73. Richardson, M.; Wallace, S. *Getting Started with Raspberry PI*; O’Reilly Media, Inc.: Sebastopol, CA, USA, 2012.

74. Vijitkunsawat, W.; Chantngarm, P. Study of the performance of machine learning algorithms for face mask detection. In *Proceedings of the 2018-5th International Conference on Information Technology (InCIT)*, Chonburi, Thailand, 21–22 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 630–645.

75. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lake City, UT, USA, 18–22 June 2018*; pp. 8697–8710.

76. Tomás, J.; ReGo, A.; Viciano-Tudela, S.; Lloret, J. Incorrect facemask-wearing detection using convolutional neural networks with transfer learning. *Healthcare* **2021**, *9*, 1050. [CrossRef]

77. Humans in the Loop. Medical Mask Dataset. 2020. Available online: https://humansintheloop.org/resources/datasets/medical-mask-dataset/ (accessed on 1 July 2022).

78. 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 2001*, Kauai, HI, USA, 8–14 December 2001; IEEE: Piscataway, NJ, USA, 2001; Volume 1, p. I.
97. Rahmatulloh, A.; Gunawan, R.; Sulastri, H.; Pratama, I.; Darmawan, I. Face Mask Detection using Haar Cascade Classifier.
98. Ruff, L.; Kauffmann, J.R.; Vandermeulen, R.A.; Montavon, G.; Samek, W.; Kloft, M.; Dietterich, T.G.; Müller, K. A Unifying
99. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
100. Boult, T.E.; Cruz, S.; Dhamija, A.R.; Günther, M.; Henrydoss, J.; Scheirer, W.J. Learning and the Unknown: Surveying Steps
101. Aggarwal, C.C.; Yu, P.S. Outlier Detection for High Dimensional Data. In Proceedings of the 2001 ACM SIGMOD International
102. Pimentel, M.A.F.; Clifton, D.A.; Clifton, L.A.; Tarassenko, L. A review of novelty detection. *Electronics* 2022, 11, 904. [CrossRef]
103. Wang, Z.; Wang, P.; Louis, P.C.; Wheless, L.E.; Huo, Y. Wearmask: Fast in-browser face mask detection with serverless edge
104. Varshini, B.; Yogesh, H.; Pasha, S.D.; Suhail, M.; Madhumitha, V.; Sasi, A. IoT-Enabled smart doors for monitoring body
105. Cheng, L.; Bui, S.; Song, Z.; Xun, F.; Wu, J.; Li, Q. Hybrid Deep Learning for Human Activity Recognition. *Adv. Intel. Inform. Sci.
106. Varshini, B.; Yogesh, H.; Pasha, S.D.; Suhail, M.; Madhumitha, V.; Sasi, A. IoT-Enabled smart doors for monitoring body
107. Liberatori, B.; Mami, C.A.; Santacatterina, G.; Zullich, M.; Pellegrino, F.A. YOLO-Based Face Mask Detection on Low-End Devices
108. Said, Y. YFnyq-YOLO-Net: An embedded quantized convolutional neural network for face mask detection in COVID-19 pandemic
109. Mohan, P.; Paul, A.J.; Chirania, A. A tiny CNN architecture for medical face mask detection for resource-constrained endpoints.
110. Fassfous, N.; Vemparala, M.R.; Frickenstein, A.; Frickenstein, L.; Badawy, M.; Stechele, W. Binarycop: Binary neural network-based
111. Liberatori, B.; Mami, C.A.; Santacatterina, G.; Zullich, M.; Pellegrino, F.A. YOLO-Based Face Mask Detection on Low-End Devices
112. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
113. Jiang, X.; Gao, T.; Nan, K.; Seong, K.; Chai, K.; Li, W.S. Real-time mask detection on google edge TPU. *arXiv* 2020, arXiv:2010.04427.
114. Said, Y. YFnyq-YOLO-Net: An embedded quantized convolutional neural network for face mask detection in COVID-19 pandemic
115. Liberatori, B.; Mami, C.A.; Santacatterina, G.; Zullich, M.; Pellegrino, F.A. YOLO-Based Face Mask Detection on Low-End Devices
116. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
117. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
118. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
119. Liberatori, B.; Mami, C.A.; Santacatterina, G.; Zullich, M.; Pellegrino, F.A. YOLO-Based Face Mask Detection on Low-End Devices
120. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
121. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
122. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
123. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
124. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
125. Ruparelia, S.; Jethva, M.; Gajjar, R. Real-time Face Mask Detection System on Edge using Deep Learning and Hardware
102. Hendrycks, D.; Gimpel, K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *arXiv* 2016, arXiv:1610.02136.
103. Liang, S.; Li, Y.; Srikan, R. Principled Detection of Out-of-Distribution Examples in Neural Networks. *arXiv* 2017, arXiv:1706.02690.
104. Denouden, T.; Salay, R.; Czarnecki, K.; Abdelzad, V.; Phan, B.; Vernekar, S. Improving Reconstruction Autoencoder Out-of-distribution Detection with Mahalanobis Distance. *arXiv* 2018, arXiv:1812.02765.
105. An, J.; Cho, S. Variational Autoencoder based Anomaly Detection Using Reconstruction Probability. *Comput. Sci.* 2015, 2015, 36663713.
106. Abdelzad, V.; Czarnecki, K.; Salay, R.; Denouden, T.; Vernekar, S.; Phan, B. Detecting Out-of-Distribution Inputs in Deep Neural Networks Using an Early-Layer Output. *arXiv* 2019, arXiv:1910.10307.
107. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* 2021, arXiv:2107.08430.
108. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 2015, 37, 1904–1916. [CrossRef]
109. Liu, W.; Wang, X.; Owens, J.; Li, Y. Energy-based out-of-distribution detection. *Adv. Neural Inf. Process. Syst.* 2020, 33, 21464–21475.
110. Zhai, S.; Cheng, Y.; Lu, W.; Zhang, Z. Deep structured energy based models for anomaly detection. In Proceedings of the International Conference on Machine Learning, PMLR, New York, NY, USA, 20–22 June 2016; pp. 1100–1109.
111. Riaz, M.; Tanveer, S.; Pamucar, D.; Qin, D.S. Topological Data Analysis with Spherical Fuzzy Soft AHP-TOPSIS for Environmental Mitigation System. *Mathematics* 2022, 10, 1826. [CrossRef]