big data and cognitive computing

MDPI

Article

Deep Learning and YOLOv8 Utilized in an Accurate Face Mask Detection System

Christine Dewi *, Danny Manongga *, Hendry, Evangs Mailoa and Kristoko Dwi Hartomo

Department of Information Technology, Satya Wacana Christian University, Salatiga 50711, Indonesia; hendry@uksw.edu (H.); evangs.mailoa@uksw.edu (E.M.); kristoko@uksw.edu (K.D.H.)
* Correspondence: christine.dewi@uksw.edu (C.D.); danny.manongga@uksw.edu (D.M.)

Abstract: Face mask detection is a technological application that employs computer vision methodologies to ascertain the presence or absence of a face mask on an individual depicted in an image or video. This technology gained significant attention and adoption during the COVID-19 pandemic, as wearing face masks became an important measure to prevent the spread of the virus. Face mask detection helps to enforce mask-wearing guidelines, which can significantly reduce the spread of respiratory illnesses, including COVID-19. Wearing masks in densely populated areas provides individuals with protection and hinders the spread of airborne particles that transmit viruses. The application of deep learning models in object recognition has shown significant progress, leading to promising outcomes in the identification and localization of objects within images. The primary aim of this study is to annotate and classify face mask entities depicted in authentic images. To mitigate the spread of COVID-19 within public settings, individuals can employ the use of face masks created from materials specifically designed for medical purposes. This study utilizes YOLOv8, a state-of-the-art object detection algorithm, to accurately detect and identify face masks. To analyze this study, we conducted an experiment in which we combined the Face Mask Dataset (FMD) and the Medical Mask Dataset (MMD) into a single dataset. The detection performance of an earlier research study using the FMD and MMD was improved by the suggested model to a "Good" level of 99.1%, up from 98.6%. Our study demonstrates that the model scheme we have provided is a reliable method for detecting faces that are obscured by medical masks. Additionally, after the completion of the study, a comparative analysis was conducted to examine the findings in conjunction with those of related research. The proposed detector demonstrated superior performance compared to previous research in terms of both accuracy and precision.

Keywords: object detection; CNN; COVID-19; face mask identification; YOLOv8; deep learning

Citation: Dewi, C.; Manongga, D.; Hendry; Mailoa, E.; Hartomo, K.D. Deep Learning and YOLOv8 Utilized in an Accurate Face Mask Detection System. *Big Data Cogn. Comput.* **2024**, *8*, 9. https://doi.org/10.3390/bdcc8010009

Academic Editor: Min Chen

Received: 28 November 2023
Revised: 6 January 2024
Accepted: 10 January 2024
Published: 16 January 2024

1. Introduction

"Face mask identification" typically refers to the process of detecting whether a person is wearing a face mask and, if so, how it is being worn (properly or incorrectly). This technology is part of the broader category of object detection in computer vision, where the object of interest is a face with or without a mask. The COVID-19 pandemic had a profound impact on the global population in the previous year, exhibiting minimal regard for demographic factors such as age, gender, or geographical boundaries. The virus caused a temporary cessation of global activities [1,2]. Due to the global outbreak of COVID-19, several countries have implemented additional regulations on the utilization of facial coverings as a preventive measure against infection. In the period preceding the onset of the COVID-19 pandemic, individuals adopted the practice of donning masks as a preventive measure against the detrimental consequences of air pollution. This behavior has persisted up to the present time [3,4].

Face mask detection technology offers several benefits, particularly in the context of public health and safety, especially during situations like the COVID-19 pandemic. Some

of the key benefits include (1) health and safety: Face mask detection helps to enforce mask-wearing guidelines, which can significantly reduce the spread of respiratory illnesses, including COVID-19. Wearing masks in crowded places helps to protect individuals and prevents the transmission of airborne particles carrying viruses [5]. (2) Disease prevention: During outbreaks of contagious diseases, like the COVID-19 pandemic, identifying individuals who are not wearing masks in public spaces can help authorities take prompt actions to mitigate the spread of the disease. This is particularly crucial in controlling outbreaks and protecting vulnerable populations [6,7]. (3) Reduced manual monitoring: Automated face mask detection reduces the need for manual monitoring and enforcement, allowing staff and authorities to focus on other tasks. This is especially important in places with many people, where manual monitoring might be challenging [8]. (4) Efficiency: automated systems can process many individuals quickly and accurately, making them well-suited for environments where efficiency is crucial, such as airports, train stations, and shopping malls [9]. (5) Consistency: automated systems provide consistent and unbiased monitoring, ensuring that all individuals are treated equally, and mask guidelines are enforced without any discrimination. (6) Alerting and reporting: Some face mask detection systems can generate alerts and reports based on the data collected. This information can be valuable for health authorities, organizations, and institutions to track compliance rates and make informed decisions. (7) Public awareness: the presence of face mask detection technology can increase public awareness of the importance of wearing masks, serving as a visual reminder for individuals to follow health guidelines [10]. (8) Adaptability: face mask detection technology can be adapted for various environments and scenarios, making it versatile for different applications, from public spaces to workplaces [11]. (9) Post-pandemic applications: while the initial focus has been on pandemic-related situations, face mask detection technology can find utility beyond the pandemic, such as in industrial safety settings or areas where respiratory protection is important [12].

Furthermore, it is imperative to carry out a study on the duration for which face masks can effectively offer protection [13,14], Additionally, endeavors should be made to extend the usability of disposable masks, while simultaneously advocating for the development and use of reusable masks. According to the World Health Organization (WHO), to effectively address and overcome the COVID-19 pandemic, governments must provide guidance and supervision to the general population in communal settings, especially in areas with high population density. The administration will achieve complete success in this conflict only upon reaching that point. The integration of surveillance systems with models of artificial intelligence, for instance, could be utilized in this case as an example of a potential application [15,16].

YOLOv8 represents the latest iteration within the YOLO series, showcasing advancements that build upon the notable attributes that contributed to the widespread acclaim of its predecessors. This is achieved through the implementation of a novel architecture that is based on transformer models, resulting in enhanced levels of precision and efficiency [17,18]. The YOLOv8 model is highly efficient at detecting objects because of its advanced training methodology, which involves the integration of knowledge distillation and pseudo-labeling techniques [19,20].

Previous studies have introduced LLE-CNNs as a method for detecting masked faces. The LLE-CNNs consist of three fundamental modules. The suggested module commences by integrating two pre-trained convolutional neural networks (CNNs) to detect probable facial regions inside the input image. These regions are subsequently described using more comprehensive descriptors, which are employed to enhance recommendations. The generation of a consistency descriptor involves the utilization of the locally linear embedding (LLE) methodology and dictionaries that have been acquired from a substantial dataset comprising generated ordinary faces, masked faces, non-faces, and other relevant methodologies. This process is carried out within the Embedding module [21]. The article referenced as [22] offers a comparative analysis of face recognition datasets that employ masked and non-masked images. Additionally, the article presents a detailed exposition of

Principal Component Analysis (PCA). They discovered statistical approaches that can be incorporated into methods for both masked and unmasked face recognition as part of their investigation. Both of these situations may benefit from the application of these methods.

This work presents a description of a facial identification model that utilizes deep transfer learning techniques for mask detection. The following are the most significant contributions made by this paper: (1) A novel deep learning detection model has been built and showcased that can automatically identify and localize a face that is wearing a medical mask within an image. (2) This study aims to identify and assess the benefits and drawbacks associated with the utilization of YOLOv8 in facial recognition systems specifically designed for the detection and recognition of medical face masks and our experiment combined the Face Mask Dataset (FMD) and Medical Mask Dataset (MMD). (3) We perform a comparative analysis of the YOLOv8s, YOLOv8n, and YOLOv8m models.

This section contains a paper outline. Section 2 describes and evaluates pertinent works that came before. Section 3 provides an overview of our proposed methodology. Section 4 presents the dataset, training data, and system test results. Section 5 concludes with additional research and development recommendations.

2. Materials and Methods

2.1. Face Mask Identification with Deep Learning

The application of deep learning algorithms and techniques in the context of face mask identification involves the detection and classification of individuals based on their adherence to wearing a face mask. Deep learning, specifically convolutional neural networks (CNNs), has demonstrated remarkable efficacy in the domain of image recognition, rendering it well-suited for the task of identifying face masks [23].

In one study, researchers employed a network based on Generative Adversarial Networks (GANs) [24]. This network consisted of two discriminators, each serving a distinct purpose. The first discriminator facilitated the acquisition of knowledge regarding the overall facial structure, while the second discriminator was afterward incorporated to specifically target the learning of intricate details within the occluded regions. In [25], the authors describe a face mask identification model that uses a hybrid approach that combines deep learning and more conventional machine learning approaches. The model that is suggested is broken up into two components. The Resnet50 feature extraction method, which is the first part of this setup, is what it is designed to be used in conjunction with, so that users may receive the most out of this tool. The authors employed the Yolo V3 algorithm for face detection, as stated in reference [26]. Moreover, the Yolo V3 model is constructed upon the Darknet-53 architecture, which functions as its foundational framework. The approach that was suggested achieved a testing accuracy of 93.9%. The major goal is to achieve proper mask identification while also decreasing the occurrence of false positive face detections to the greatest extent practicable. This will ensure that warnings are only activated for medical personnel who are not performing their tasks while wearing a surgical mask. The proposed system was accurate to a 95% level throughout the archiving process.

Another research study (Ejaz et al., 2019) [22] implements the Principal Component Analysis (PCA) algorithm with the Olivetti Research Laboratory (ORL) face dataset and achieves 70% accuracy. In their work, a statistical procedure is selected, which is applied in non-masked face recognition and also applied in the masked face recognition technique. PCA is a more effective and successful statistical technique and is widely used.

Moreover, Ge et al., 2017 [21] employ A Dataset of Masked Faces (MAFA). The author suggests using LLE-CNNs for detecting masked faces based on the dataset. LLE-CNNs have three main modules. The Proposal module initially merges two pre-trained convolutional neural networks (CNNs) to identify potential facial regions within the input image and encode them using descriptors with many dimensions. Subsequently, the Embedding module is integrated to transform these descriptors into a similarity-based descriptor through the use of the locally linear embedding (LLE) technique and the dictionaries

trained on a substantial collection of synthesized normal faces, masked faces, and non-faces. By employing this approach, a significant portion of the absent facial signals can be effectively restored, and the negative effects caused by various masks that introduce distorted cues can be significantly reduced.

2.2. YOLOv8 Architecture

The core structure of YOLOv8 is quite like that of YOLOv5, except for the C3 module, which has been replaced with the C2f module. This module is derived from the CSP idea. YOLOv8's C2f module was produced by using the ELAN concept from YOLOv7 and combining it with C3. This was carried out in order to develop the module. This integration was undertaken to improve YOLOv8's gradient flow information without jeopardizing its lightweight design in any way [27]. The dominant SPPF module was used throughout the entirety of the final stage of the backbone architecture. After this, a sequential application of three waxpools, each of which had a size of 5 by 5 inches, was carried out. After that, the output of each layer was concatenated to ensure the accurate detection of objects at various scales while keeping a lightweight design. This was accomplished without sacrificing accuracy [28].

Bounding boxes in the form of annotations are widely prevalent in the field of deep learning, surpassing other types of annotations in terms of frequency [29]. Within the domain of computer vision, the term "bounding boxes" refers to rectangular shapes utilized to delineate and specify the precise spatial coordinates of the object under scrutiny. The coordinates located at both the upper-left and lower-right corners of the rectangle can be utilized to determine their position regarding the x and y axes. The upper-left corner of the rectangle and the lower-right corner of the rectangle both contain these coordinates. In the context of activities involving object detection and localization, bounding boxes are used quite frequently. To generate a bounding box for each sign, the BBox label tool [30] is employed.

A total of three distinct kinds of labels—numbered 0, 1, and 2—must be applied to complete the labeling process. In contrast to the input formats used by other programs, YOLO does not use object coordinates to express the data in its input values. The coordinates of the object's center point are included in the YOLO input data alongside the dimensions of the object's width and height (x, y, w, h). Common methods of representing bounding boxes include the use of either two coordinates, ($x1$, $y1$) and ($x2$, $y2$), or a single coordinate, ($x1$, $y1$), in conjunction with the width (w) and height (h) of the bounding box. The process of transformation is depicted using Equations (1)–(6).

$$dw = 1/w \tag{1}$$

$$x = \frac{x1 + x2}{2} \times dw \tag{2}$$

$$dh = 1/h \tag{3}$$

$$y = \frac{y1 + y2}{2} \ dh \tag{4}$$

$$w = (x2 - x1) \times dw \tag{5}$$

$$h = (y2 - y1) \times dh \tag{6}$$

In these equations, w represents the width of the image, while h represents its height. Next, we predict the width (w) and height (h) of the boxes x, y, and anchor box (dw and dh). LabelImg is a software application used for visually marking and identifying objects within images [31]. The program is implemented in the Python programming language and utilizes the Qt framework for its graphical user interface. Annotations are stored as XML files in the PASCAL VOC format, which is the same format utilized by ImageNet. In addition, it also provides support for YOLO and CreateML formats. We used the LabelImg to label our images and a corresponding text file with the same name as each

image file in the same directory will be generated. Each text file includes the object class, object coordinates, the height and width of the associated picture file, as well as additional metadata. Convolution, batch normalization, and SiLu activation functions for the YOLOv8 architecture are the three basic components that make up the convolutional neural network (CNN) that are depicted in Figure 1.

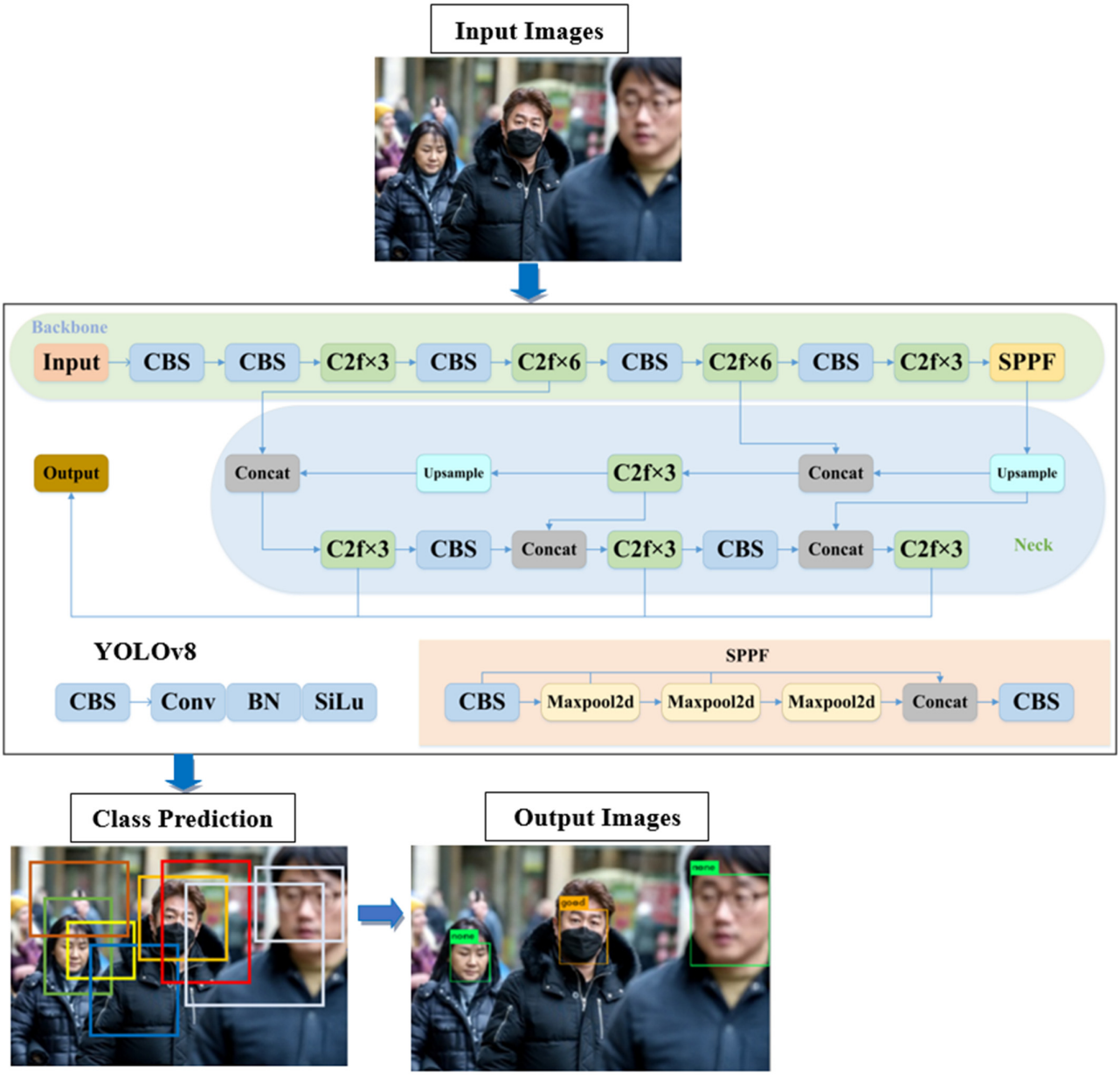


Figure 1. System architecture of YOLOv8.

The loss function used in YOLO is a combination of localization loss and classification loss. The purpose of this loss function is to measure the difference between the predicted bounding boxes and class probabilities and the ground-truth annotations. The components of the YOLO loss are as follows: (1) localization loss: The localization loss evaluates the accuracy of the predicted bounding box locations. It is often represented using metrics like MSE or MAE between the predicted box coordinates (center coordinates, width, and height), and the ground-truth box coordinates. (2) Confidence loss: The confidence loss measures how well the model predicts the confidence score for each bounding box. It is computed as the intersection over union (IoU) between the predicted box and the ground-truth box. The loss could be the binary cross-entropy loss between the predicted confidence

score and the ground-truth indicator (whether an object is present or not). (3) Classification loss: The classification loss quantifies the precision with which the model predicts the class probabilities for each bounding box. The determination of this is achieved through the computation of the cross-entropy loss, which involves comparing the predicted class probabilities with the actual class labels represented as one-hot vectors. The overall YOLO loss value is a weighted sum of these individual loss components. The exact formulation can be different between YOLO versions, and some versions might include additional terms or regularization components. Below is the YOLO loss function in Equation (7) [32].

Yolo Loss Function

$$\begin{aligned} &= \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y - \hat{y}_i)^2 \right] \\ &+ \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] + \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 \\ &+ \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 + \sum_{i=0}^{s^2} \mathbb{I}_{i}^{obj} \sum_{c \epsilon classes} \left(p_i(\mathrm{c}) - \hat{p}_i(c) \right)^2 \end{aligned} \tag{7}$$

where $\mathbb{I}_{ij}^{obj}$ indicates whether the object appears in cell i, and $\mathbb{I}_{ij}^{obj}$ denotes that the jth bounding box predictor in cell i is responsible for the prediction. Next, $\left(\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{c}, \hat{p} \right)$ is implemented to express the anticipated bounding box's center coordinates, width, height, confidence, and category probability. This experiment employed the $\lambda coord$ to 0.5, demonstrating that the width and height errors are less useful in the computation. To mitigate the effect of numerous vacant grids on the loss value, $\lambda noobj$ = 0.5 is utilized.

In object detection, there can be multiple classes of objects to detect. Next, the mAP calculates the AP for each class and then computes the mean of these AP values. This provides an overall assessment of the model's performance across all classes, accounting for the varying difficulty levels of detecting different objects. The *mAP* is described in Equation (8).

$$mAP = \int_0^1 p(o) do \tag{8}$$

The variable $p(o)$ represents the precision of the object detection. The intersection over union (*IoU*) metric quantifies the degree of overlap between the bounding boxes of the prediction (*pred*) and the ground truth (*gt*), as expressed in Equation (9). Precision and recall are represented based on [33] in Equations (10) and (11).

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} = TP/N \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

where *TP* represents true positives, *FP* represents false positives, *FN* represents false negatives, and *N* represents the total number of objects recovered, including the true positives and false positives. Another evaluation index, *F*1 [34], is shown in Equation (12).

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

3. Results

3.1. Face Mask Dataset (FMD) and Medical Mask Dataset (MMD)

The research conducted in this study utilized two distinct datasets of medical face masks, which were publicly accessible. First, the Face Mask Dataset (FMD) in [35] is a

publicly available masked face dataset. The FMD dataset consists of 853 pictures, which are stored in the PASCAL VOC format. The images shown in Figure 2 are samples of the FMD. Following that, the Medical Mask Dataset (MMD) may be found on Kaggle [36]. In addition, the MMD collection comprises 682 images, each of which comprises more than 3000 disguised faces that have been medically masked. In this investigation, all participants gave their informed consent, and Figure 2 provides some samples of pictures in MMD.

Figure 2. Sample images in the experimental Face Mask Dataset (FMD), and Medical Mask Dataset (MMD).

The experimental setup involved merging the MMD and FMD datasets to create a distinct and extensive dataset. A dataset was utilized to gather a total of 1415 pictures, which underwent a rigorous curation process involving the removal of low-quality images and duplicates from the original dataset.

Figure 3 illustrates the incorporation of the Medical Mask Dataset (MMD) and Face Mask Dataset (FMD) into our study. The MMD consists of three unique categories, specifically bad, good, and none. In contrast, the FMD comprises three distinct classifications denoted as mask_weared_incorrect, with_mask, and without_mask. The experiment explains the three categories in the following manner: the category labeled as "bad" corresponds to instances where masks were worn incorrectly, the category labeled as "good" corresponds to instances where masks were worn properly, and the category labeled as "none" corresponds to instances where masks were not worn at all [37]. The class categorized as "bad" consists of approximately 500 instances, whereas the class categorized as "good" encompasses more than 4000 occurrences. The "none" class, on the other hand, consists of over 500 instances. The range of x and y values is from 0.0 to 1.0, while the width varies from 0.0 to 0.6, and the height is from 0.0 to 0.8. In the absence of immunization, masks are one of the few preventative measures that can be taken against COVID-19; as a result, they play an important part in preserving patients' respiratory health and preventing the spread of respiratory illnesses. The YOLO format entails the presence of a corresponding text file with a .txt extension for each JPEG image file. The provided text file contains

comprehensive data regarding the position of each item inside the image, encompassing its class and x and y coordinates, as well as its width and height.

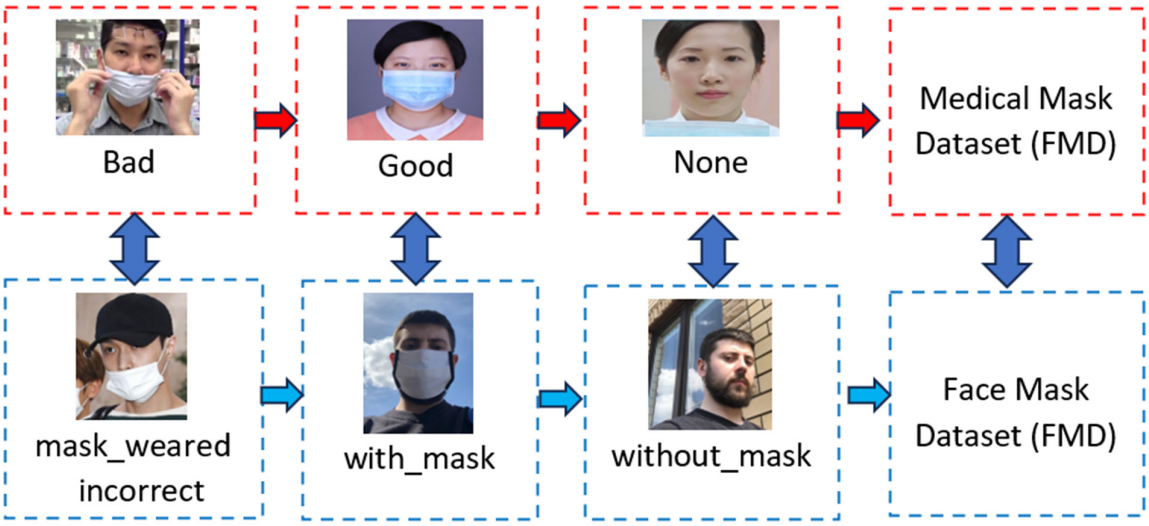


Figure 3. The combination of MMD and FMD datasets.

3.2. Training Result

Data augmentation is an approach that is frequently utilized in machine learning and deep learning to artificially increase the variety of a training dataset by performing various changes to the original data. By presenting a machine learning model with a wider variety of training data, the generalization and resiliency of the model can be improved with data augmentation. During the training process, we will make use of a variety of techniques for augmenting the data, such as padding, cropping, and horizontal flipping, among others. These techniques are widely employed in the construction of large neural networks because of their advantageous qualities. During the training process, we employed 100 epochs, weight_decay = 0.0005, learning rate = 0.001, batch size = 16, an image size of 416, and an IoU threshold value of 0.5. The model converged and achieved satisfactory performance in our experiments so that there was no significant increase in performance after the 100th epoch.

In addition, the training model environment comprised an Nvidia RTX3080Ti GPU accelerator with 11 gigabytes of RAM, an i7 central processing unit (CPU), and 16 gigabytes of DDR2 memory. Training for the YOLOv8 was carried out on a single graphics processing unit (GPU), and one of its primary goals was the achievement of real-time detection. While the remaining thirty percent of the information was used for testing reasons, the remaining seventy percent was utilized for training purposes. The YOLO algorithm is designed to forecast various bounding boxes within each grid cell. During the training phase, it is desirable to assign the responsibility of predicting the bounding box for each item to only one predictor. The YOLO algorithm designates a single predictor as the "responsible" entity for object prediction, determined by selecting the prediction with the highest current intersection over union (IOU) value for the ground truth. This phenomenon results in a specialization among the bounding box predictors. The performance of each predictor in predicting specific sizes, aspect ratios, or types of objects leads to an enhancement in the overall recall score.

Figure 4 provides a visual representation of the steps involved in the training process for the test batch with 0 labels and the test batch with 0 predictions.

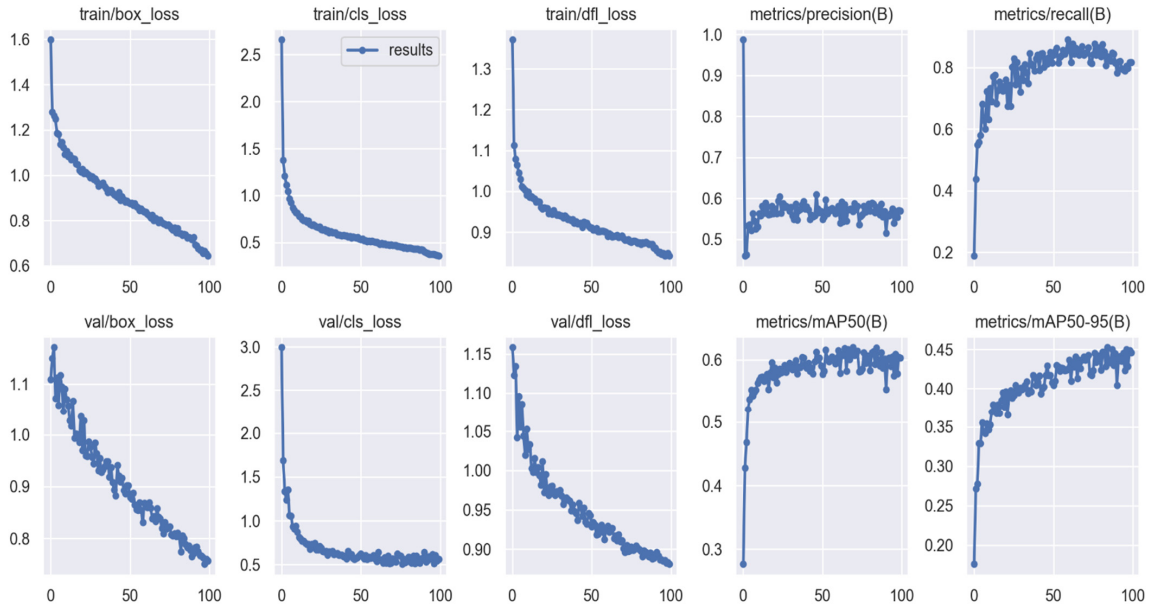
(**a**)

(**b**)

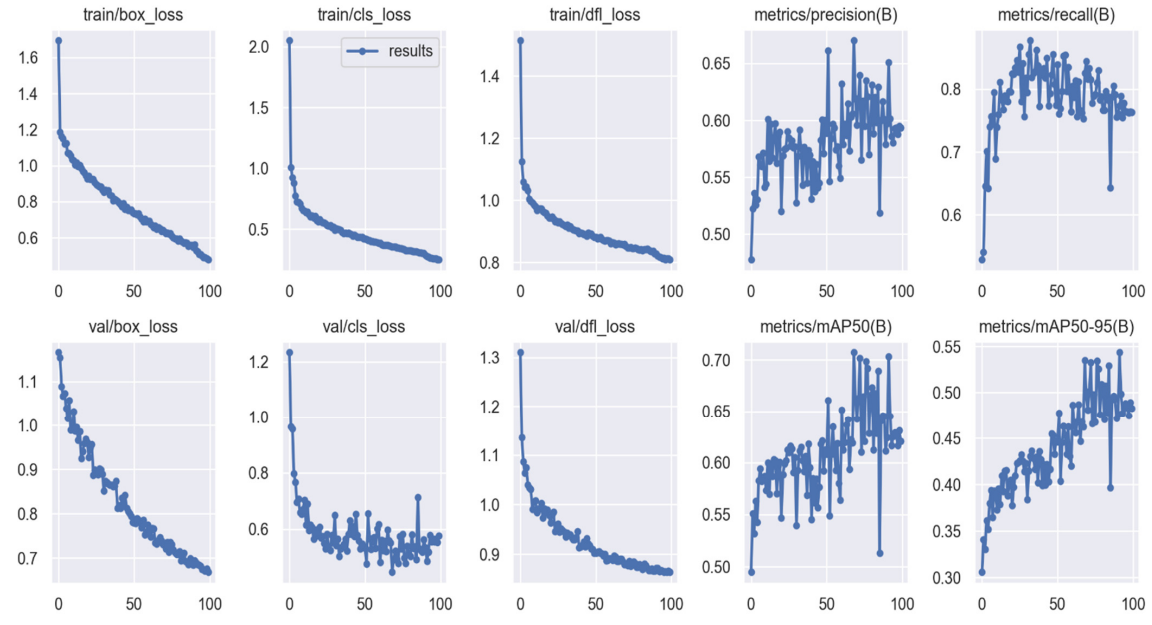
Figure 4. Training process for the (**a**) test batch with 0 labels and (**b**) the test batch with 0 predictions. The human features depicted in the figures were obtained from publicly available datasets (FMD and MMD).

Figure 5 illustrates the training graph of YOLOv8n, YOLOv8s, and YOLOv8m over 100 epochs. Non-maximum suppression (NMS) is a crucial technique utilized by YOLO models. During training, the YOLOv8m loss values are as follows: box_loss = 0.38, cls_loss = 0.19, and dfl_loss = 0.79. The YOLOv8n loss values are box_loss = 0.64, cls_loss = 0.35, and dfl_loss = 0.84. The YOLOv8s loss values are box_loss = 0.47, cls_loss = 0.25, and dfl_loss = 0.80. The model reached convergence and demonstrated satisfactory performance in our experiments, with no notable improvement in performance beyond the 100th epoch. NMS improves object detection after processing. Multiple bounding boxes are often generated for an image object during object detection. Although these bounding boxes may overlap or

be in various locations, they all represent the same object. Loss curves depict the temporal evolution of a model's performance, specifically the number of iterations or steps executed by the model. They assist in determining the adequacy of our model in fitting the data, avoiding overfitting or underfitting, and diagnosing the representativeness of the datasets and/or the number of training steps. The term "box_loss" refers to the loss function used for bounding box regression. It quantifies the discrepancy between the predicted bounding box coordinates and dimensions and the corresponding ground-truth values. A decrease in box loss indicates an improvement in the precision of the predicted bounding boxes. The term "cls_loss" refers to the classification loss, which quantifies the discrepancy between the predicted class probabilities for each object in the image and the corresponding ground-truth values. A decrease in the cls_loss metric indicates that the model is exhibiting improved accuracy in predicting the class of the objects.

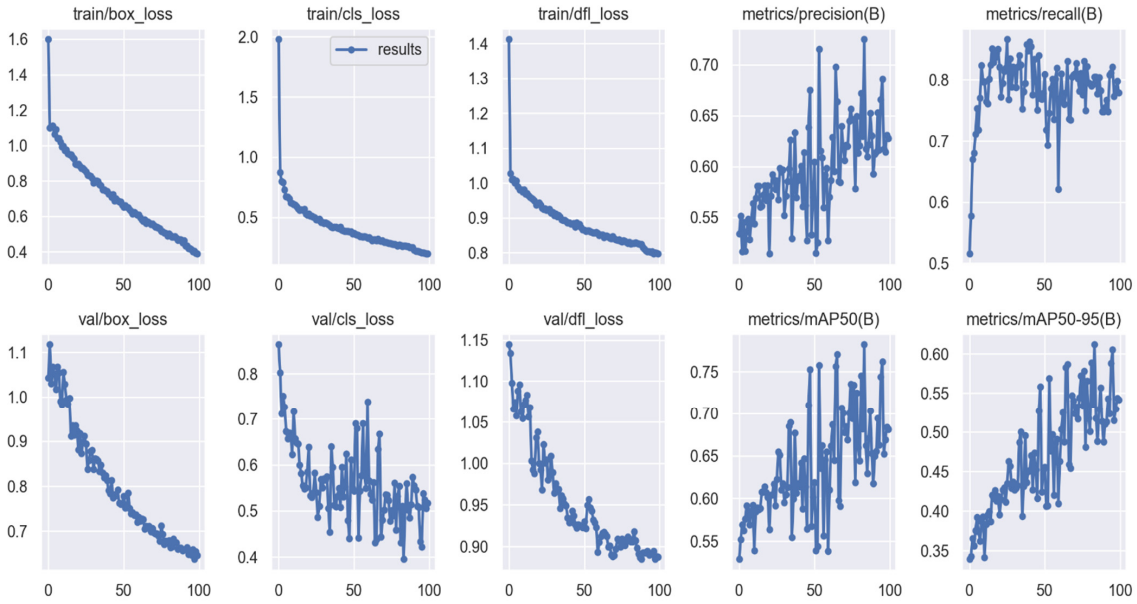


(**a**)



(**b**)

Figure 5. *Cont.*



(**c**)

Figure 5. Training performance using (**a**) YOLOv8n, (**b**) YOLOv8s, and (**c**) YOLOv8m.

The "dfl_loss" refers to the loss function associated with the deformable convolution layer, which is a novel component incorporated into the YOLO design in YOLOv8. The loss function quantifies the discrepancy in the deformable convolution layers, which are specifically engineered to enhance the model's capability of detecting objects with diverse sizes and aspect ratios. A decreased dfl_loss value suggests that the model exhibits more proficiency in managing object deformations and variations in appearance. The aggregate loss value is commonly computed as a combination of these individual losses, with each loss being assigned a specific weight. The units on the vertical axis may vary depending on the individual implementation, but in general, they indicate the size of the error or discrepancy between the predicted values and the ground-truth values. Figure 5 demonstrates that our findings do not exhibit signs of overfitting or underfitting.

Table 1 displays the performance of all classes after the training session. The table shown above offers a comprehensive overview of many performance measures for the evaluation of a model's performance. These metrics include the training loss value, mean average precision (mAP), average precision (AP), precision, recall, F1 score, and intersection over union (IoU) performance. The metrics are provided for each distinct class. The YOLOv8m model obtains a maximum average mean average precision (mAP) of 78.4%. Additionally, it achieves an mAP of 99.1% specifically for the "good" class. Subsequently, YOLOv8s demonstrates a mean average precision (mAP) of 70.4% for the "good" class, while achieving a remarkable 99% mAP overall.

Table 1. FMD and MMD training performance for all models.

Model	Class	Images	Labels	P	R	mAP@.5
YOLO V5s	all	507	2661	0.639	0.832	0.662
	bad	507	260	0.508	0.815	0.492
	good	507	2123	0.933	0.945	0.963
	none	507	278	0.476	0.736	0.496
YOLO V5m	all	507	2661	0.639	0.832	0.672
	bad	507	260	0.508	0.815	0.492
	good	507	2123	0.933	0.945	0.964
	none	507	278	0.476	0.736	0.496

Table 1. *Cont.*

Model	Class	Images	Labels	P	R	mAP@.5
YOLO7x	all	456	2154	0.625	0.932	0.635
	bad	456	72	0.612	0.944	0.163
	good	456	1733	0.942	0.965	0.977
	none	456	349	0.772	0.885	0.494
YOLO7	all	456	2154	0.609	0.931	0.632
	bad	456	72	0.166	0.941	0.168
	good	456	1733	0.939	0.978	0.986
	none	456	349	0.722	0.874	0.742
YOLOv8n	all	456	2154	0.579	0.834	0.619
	bad	456	72	0.142	0.687	0.152
	good	456	1733	0.891	0.969	0.982
	none	456	349	0.702	0.845	0.723
YOLOv8s	all	456	2154	0.651	0.774	0.704
	bad	456	72	0.217	0.681	0.367
	good	456	1733	0.968	0.975	0.99
	none	456	349	0.768	0.754	0.754
YOLOv8m	all	456	2154	0.716	0.829	0.784
	bad	456	72	0.355	0.681	0.5
	good	456	1733	0.983	0.972	0.991
	none	456	349	0.811	0.834	0.861

4. Discussions

The testing accuracy for all classes of the MMD and FMD can be found in Table 2. These classes include bad, good, and none. According to the obtained test results, it was observed that YOLOv8m achieved the highest mean average precision (mAP) of 78.4% for the category "good" when compared to the other models utilized in the conducted experiment. Class ID 1, denoted as "good", exhibited the highest average accuracy, achieving around 99.1%. This was followed by Class ID 0, referred to as "bad", which achieved a comparatively lower accuracy of 50%. Class ID 2, labeled as "none", achieved an average accuracy of 86.1%.

The outcomes of the MMD and FMD recognition using YOLOV8m are depicted in Figure 6. The model we propose has a high level of accuracy at detecting items inside an image. YOLOV8m is capable of discerning between different classes, namely "bad", "good", and "none", based on the presence of either a single object or multiple objects within a given image. The YOLOV8m model demonstrates better performance in medical-masked face identification, surpassing competing models. The efficiency of the suggested model in identifying face masks has been successfully introduced.

Table 2. Testing the models' accuracy with FMD and MMD.

Model	Class	Images	Labels	P	R	mAP@.5
YOLO V5s	all	507	2661	0.615	0.837	0.662
	bad	507	260	0.475	0.777	0.48
	good	507	2123	0.932	0.958	0.97
	none	507	278	0.471	0.791	0.522
YOLO V5m	all	507	2661	0.626	0.886	0.671
	bad	507	260	0.481	0.858	0.523
	good	507	2123	0.931	0.957	0.972
	none	507	278	0.465	0.845	0.509
YOLO7x	all	456	2154	0.625	0.932	0.635
	bad	456	72	0.612	0.944	0.163
	good	456	1733	0.942	0.965	0.977
	none	456	349	0.772	0.885	0.494

Table 2. *Cont.*

Model	Class	Images	Labels	P	R	mAP@.5
YOLO7	all	456	2154	0.609	0.931	0.632
	bad	456	72	0.166	0.941	0.168
	good	456	1733	0.939	0.978	0.986
	none	456	349	0.722	0.874	0.742
YOLOv8n	all	456	2154	0.579	0.839	0.619
	bad	456	72	0.148	0.712	0.152
	good	456	1733	0.893	0.969	0.981
	none	456	349	0.695	0.836	0.725
YOLOv8s	all	456	2154	0.652	0.773	0.706
	bad	456	72	0.22	0.681	0.367
	good	456	1733	0.965	0.975	0.99
	none	456	349	0.77	0.665	0.762
YOLOv8m	all	456	2154	0.716	0.829	0.784
	bad	456	72	0.355	0.681	0.5
	good	456	1733	0.983	0.972	0.991
	none	456	349	0.811	0.834	0.861

Table 3 displays the results of the tests that compared the CNN models concerning GFLOPS, parameters, and layers. Loading 168 layers and 28.4 and 8.1 GFLOPs, respectively, is accomplished by the YOLOv8n and YOLOv8s. During training, YOLOv8m utilizes a total of 218 layers and 25,841,497 parameters.

Table 3. A summary of YOLOv8 models utilizing the FMD and MMD.

Process	Model	Layers	Parameters	GFLOPs	Speed (ms)	Inference (ms)	Post-Process Per Image
Train	Yolov8s	168	11,126,745	28.4	0.7	85.1	0.3
Train	Yolov8n	168	3,006,233	8.1	0.6	36.6	0.3
Train	Yolov8m	218	25,841,497	78.227	0.6	178	0.3
Val	Yolov8s	168	11,126,745	28.4	1	131.8	0.5
Val	Yolov8n	168	3,006,233	8.1	1	62.2	0.5
Val	Yolov8m	218	25,841,497	78.7	0.7	210.1	0.3

(**a**)

Figure 6. *Cont.*

(**b**)

Figure 6. Recognition results using YOLOv8n: (**a**) good class and (**b**) good and none.

A comparison to the preceding study is described in Table 4. Ejaz et al. (2019) [22] proposed a PCA and exhibited only a 70% mAP in an experiment with the Olivetti Research Laboratory (ORL) face dataset. Another researcher (Ge et al., 2017) [21] implemented LLE-CNN and achieved a 76.4% mAP with A Dataset of Masked Faces (MAFA). Our proposed YOLOv8n method, with 100 epochs, outperforms prior models on the FMD and MMD in terms of the mAP, with an accuracy of 78.4%. We were able to boost the overall performance of a recent study on face mask detection in this research. Furthermore, Dewi et al. (2023) [8] implemented YOLOv5m and achieved a 67.1% mAP. Next, Dewi et al. (2023) [9] proposed YOLOv7, with a 63.2% mAP. For the good class, our model, YOLOv8m, achieved the highest accuracy (99.1%) compared to other models in the experiment with the FMD and MMD.

Table 4. Previous research comparison.

Reference	Dataset	Methodology	Classification	Detection	Result AP (%)
Dewi et al., 2023 [8]	FMD and MMD	YOLOv5m	Yes	Yes	All: 67.1%, bad: 52.3%, good: 97.2%, none: 50.9%
Dewi et al., 2023 [9]	FMD and MMD	YOLOv7	Yes	Yes	All: 63.2%, bad: 16.8%, good: 98.6%, none: 74.2%
Proposed Method	FMD and MMD	YOLOv8m	Yes	Yes	All: 78.4%, bad: 50%, good: 99.1%, none: 86.1%

The advantages of YOLOv8 are numerous and varied, and the following is a non-exhaustive list of some of them: One of YOLOv8's main advantages over competing deep learning architectures is the impressive speed it delivers. Ultralytics claims that the YOLOv8 model's significant improvement in picture segmentation yields a staggering throughput of 81 frames per second. When compared to other sophisticated models like Mask R-CNN, which can only process about six frames per second, this one performs exceptionally well. Autonomous vehicles, surveillance systems, and video analytics are all examples of real-time applications where processing speed is of the utmost significance. Further, YOLOv8 can quickly detect objects and segments in an image while maintaining a high level of precision in its analysis of those elements. Because they reduce the number of false positives and negatives, the updated loss function and cutting-edge architecture of

the model each contribute to the model's improved accuracy. Moreover, because YOLOv8 provides a unified architecture for training models, it is now possible to carry out a wide variety of image segmentation tasks with just a single model. This was not possible in previous versions of the software. Object recognition, instance segmentation, and image classification are some of the tasks that fall under this category. This allows for flexibility to be realized. This versatility is vital for applications that need to complete a variety of activities, such as video surveillance and image search engines. These applications require the execution of a variety of tasks. One other illustration is the use of self-driving vehicles.

The limitations of YOLOv8 are based on its potential issues: (1) YOLO models, including YOLOv8, can struggle with detecting small objects accurately. The model may have limitations in representing fine details, leading to misidentification or low confidence scores for smaller objects. (2) Like many object detection models, YOLOv8 might face challenges when objects are partially occluded. If an object is obscured by another, the model may fail to detect it accurately. (3) YOLOv8's performance is heavily reliant on the quality, diversity, and representativeness of the training data. Inadequate or biased training data can result in poor generalization of real-world scenarios. (4) Changes in lighting conditions, variations in background, or different environmental factors may impact YOLOv8's performance. The model may not generalize well to scenes that differ significantly from the training data. (5) YOLOv8 may struggle with class imbalances in the dataset. If certain classes are under-represented, the model might be less accurate in detecting objects from those classes. (6) Depending on the application domain, YOLOv8 may require fine-tuning or adaptation to perform optimally. Using a model that is not tailored to the specific characteristics of the target domain may lead to misidentification. YOLOv8 has several hyperparameters, and the model's performance can be sensitive to their values. Selecting appropriate hyperparameters is crucial for achieving optimal results.

5. Conclusions

This article presents and discusses the findings of research into CNN-based object identification algorithms, specifically YOLOv8n, YOLOv8s, and YOLOv8m. These algorithms were developed by YOLO Labs. According to the findings of our tests, the level of precision offered by YOLOv8m is superior to anything else that is now available. Within the scope of this research, we provide an original model for medical-masked face recognition that places primary emphasis on the medical mask object. The transmission of the COVID-19 virus from one individual to another is something that should be avoided at all costs using this paradigm. In addition, the YOLOv8m technique that we have advocated, with 100 epochs, beats earlier models that have been applied to the FMD and MMD in terms of the mAP. With an average accuracy of 78.4% and a good classification, obtaining a 99.1% mAP, this technique exceeds earlier models that have been used for the FMD and MMD. This is because the YOLOv8m technique achieves a higher level of accuracy than the earlier models. We have demonstrated that the YOLOv8m model scheme that we have suggested is an effective model for detecting medical face masks. Additionally, future studies will investigate the feasibility of using deep learning models to recognize a subset of disguised faces in both still images and video. In addition, we plan to investigate how medical mask identification might benefit from the application of explainable artificial intelligence (XAI).

Author Contributions: Conceptualization, C.D.; data curation, C.D., E.M. and K.D.H.; formal analysis, D.M. and E.M.; funding acquisition, K.D.H.; investigation, H.; methodology, C.D., H. and E.M.; resources, D.M.; software, C.D.; supervision, D.M.; validation, C.D.; visualization, H.; writing—original draft, C.D.; writing—review and editing, C.D. and K.D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Vice-Rector of Research, Innovation and Entrepreneurship at Satya Wacana Christian University (078/SPK-PF/RIK/8/2023).

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the reason that we use a public and free dataset from Kaggle (FMD) and the MMD.

Informed Consent Statement: Informed consent was waived for this study due to the reason that we use the public and free dataset from Kaggle (FMD) and the MMD, and that the figures with human faces are from these public datasets.

Data Availability Statement: FMD (https://www.kaggle.com/datasets/andrewmvd/face-mask-detection) (accessed on 13 January 2023), MMD (https://www.kaggle.com/datasets/shreyashwaghe/medical-mask-dataset) (accessed on 15 January 2023), and FMD + MMD (https://drive.google.com/drive/folders/1V2UW5jLJ1uMnSIUlwIBVthS-v_OKBuYo?usp=sharing) (accessed on 17 January 2023).

Acknowledgments: The authors would like to thank all colleagues from Satya Wacana Christian University, Indonesia, and all involved in this research.

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

References

1. 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]
2. Cao, Z.; Shao, M.; Xu, L.; Mu, S.; Qu, H. Maskhunter: Real-Time Object Detection of Face Masks during the COVID-19 Pandemic. *IET Image Process.* **2020**, *14*, 4359–4367. [CrossRef]
3. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. Fighting against COVID-19: A Novel Deep Learning Model Based on YOLO-v2 with ResNet-50 for Medical Face Mask Detection. *Sustain. Cities Soc.* **2021**, *65*, 102600. [CrossRef] [PubMed]
4. Wei, Z.; Chang, M.; Zhong, Y. Fruit Freshness Detection Based on YOLOv8 and SE Attention Mechanism. *Acad. J. Sci. Technol.* **2023**, *6*, 195–197. [CrossRef]
5. 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]
6. Wang, Z.; Wang, P.; Louis, P.C.; Wheless, L.E.; Huo, Y. WearMask: Fast in-Browser Face Mask Detection with Serverless Edge Computing for COVID-19. *Electron. Imaging* **2023**, *35*, HPCI-229. [CrossRef]
7. Eyiokur, F.I.; Kantarcı, A.; Erakın, M.E.; Damer, N.; Ofli, F.; Imran, M.; Križaj, J.; Salah, A.A.; Waibel, A.; Štruc, V.; et al. A Survey on Computer Vision Based Human Analysis in the COVID-19 Era. *Image Vis. Comput.* **2023**, *130*, 104610. [CrossRef]
8. Dewi, C.; Christanto, H.J. Automatic Medical Face Mask Recognition for COVID-19 Mitigation: Utilizing YOLO V5 Object Detection. *Rev. D'intelligence Artif.* **2023**, *37*, 627–638. [CrossRef]
9. Dewi, C.; Shun Chen, A.P.; Juli Christanto, H. YOLOv7 for Face Mask Identification Based on Deep Learning. In Proceedings of the 2023 15th International Conference on Computer and Automation Engineering (ICCAE), Sydney, Australia, 3–5 March 2023; IEEE: Piscataway, NJ, USA; pp. 193–197.
10. Alsalamah, M.S.I. Automatic Face Mask Identification in Saudi Smart Cities: Using Technology to Prevent the Spread of COVID-19. *Inf. Sci. Lett.* **2023**, *12*, 2411–2422. [CrossRef]
11. Maradey-Lázaro, J.G.; Rincón-Quintero, A.D.; Garnica, J.C.R.; Segura-Caballero, D.O.; Sandoval-Rodríguez, C.L. Development of a Monitoring System for COVID-19 Monitoring in Early Stages. *Period. Eng. Nat. Sci.* **2023**, *11*, 48–61. [CrossRef]
12. Wu, X.; Sahoo, D.; Hoi, S.C.H. Recent Advances in Deep Learning for Object Detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
13. Singh, S.; Ahuja, U.; Kumar, M.; Kumar, K.; Sachdeva, M. Face Mask Detection Using YOLOv3 and Faster R-CNN Models: COVID-19 Environment. *Multimed. Tools Appl.* **2021**, *80*, 19753–19768. [CrossRef]
14. Mazen, F.M.A.; Seoud, R.A.A.; Shaker, Y.O. Deep Learning for Automatic Defect Detection in PV Modules Using Electroluminescence Images. *IEEE Access* **2023**, *11*, 57783–57795. [CrossRef]
15. Lemke, M.K.; Apostolopoulos, Y.; Sönmez, S. Syndemic Frameworks to Understand the Effects of COVID-19 on Commercial Driver Stress, Health, and Safety. *J. Transp. Health* **2020**, *18*, 100877. [CrossRef]
16. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* **2023**, *7*, 304. [CrossRef]
17. Kang, J.; Zhao, L.; Wang, K.; Zhang, K. Research on an Improved YOLOV8 Image Segmentation Model for Crop Pests. *Adv. Comput. Signals Syst.* **2023**, *7*, 1–8.
18. Patel, S.H.; Kamdar, D. Accurate Ball Detection in Field Hockey Videos Using YOLOV8 Algorithm. *Int. J. Adv. Res. Ideas Innov. Technol.* **2023**, *9*, 411–418.
19. Wang, G.; Chen, Y.; An, P.; Hong, H.; Hu, J.; Huang, T. UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors* **2023**, *23*, 7190. [CrossRef]
20. Ultralytics YOLOv8. Available online: https://github.com/ultralytics/ultralytics?ref=blog.roboflow.com (accessed on 12 May 2023).
21. Ge, S.; Li, J.; Ye, Q.; Luo, Z. Detecting Masked Faces in the Wild with LLE-CNNs. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2016; Volume 2017.

22. Ejaz, M.S.; Islam, M.R.; Sifatullah, M.; Sarker, A. Implementation of Principal Component Analysis on Masked and Non-Masked Face Recognition. In Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT 2019, Dhaka, Bangladesh, 3–5 May 2019.
23. Chen, W.; Gao, L.; Li, X.; Shen, W. Lightweight Convolutional Neural Network with Knowledge Distillation for Cervical Cells Classification. *Biomed. Signal Process. Control* **2022**, *71*, 103177. [CrossRef]
24. Dewi, C.; Chen, R.; Liu, Y.; Yu, H. Various Generative Adversarial Networks Model for Synthetic Prohibitory Sign Image Generation. *Appl. Sci.* **2021**, *11*, 2913. [CrossRef]
25. Loey, M.; Manogaran, G.; Taha, M.H.N.; Khalifa, N.E.M. A Hybrid Deep Transfer Learning Model with Machine Learning Methods for Face Mask Detection in the Era of the COVID-19 Pandemic. *Meas. J. Int. Meas. Confed.* **2021**, *167*, 108288. [CrossRef] [PubMed]
26. Nieto-Rodríguez, A.; Mucientes, M.; Brea, V.M. System for Medical Mask Detection in the Operating Room through Facial Attributes. In *Pattern Recognition and Image Analysis, Proceedings of the 2015 Iberian Conference on Pattern Recognition and Image Analysis, Santiago de Compostela, Spain, 17–19 June 2015*; Paredes, R., Cardoso, J., Pardo, X., Eds.; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2015; Volume 9117.
27. Khalid, S.; Oqaibi, H.M.; Aqib, M.; Hafeez, Y. Small Pests Detection in Field Crops Using Deep Learning Object Detection. *Sustainability* **2023**, *15*, 6815. [CrossRef]
28. Yang, W.; Wu, J.; Zhang, J.; Gao, K.; Du, R.; Wu, Z.; Firkat, E.; Li, D. Deformable Convolution and Coordinate Attention for Fast Cattle Detection. *Comput. Electron. Agric.* **2023**, *211*, 108006. [CrossRef]
29. Sharma, N.; Baral, S.; Paing, M.P.; Chawuthai, R. Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms. *Sensors* **2023**, *23*, 5843. [CrossRef]
30. Bbox Label Tool. Available online: https://github.com/puzzledqs/BBox-Label-Tool (accessed on 9 January 2024).
31. Tzutalin Labellmg. Available online: https://github.com/csq20081052/labelImg (accessed on 13 January 2022).
32. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
33. Yuan, Y.; Xiong, Z.; Wang, Q. An Incremental Framework for Video-Based Traffic Sign Detection, Tracking, and Recognition. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1918–1929. [CrossRef]
34. Kang, H.; Chen, C. Fast Implementation of Real-Time Fruit Detection in Apple Orchards Using Deep Learning. *Comput. Electron. Agric.* **2020**, *168*, 105108. [CrossRef]
35. Larxel Face Mask Detection. Available online: https://www.kaggle.com/datasets/andrewmvd/face-mask-detection (accessed on 9 January 2024).
36. Mikolaj Witkowski Medical Mask Dataset. Available online: https://www.kaggle.com/datasets/mloey1/medical-face-mask-detection-dataset (accessed on 9 January 2024).
37. Chen, R.-C.; Zhuang, Y.-C.; Chen, J.-K.; Dewi, C. Deep Learning for Automatic Road Marking Detection with Yolov5. In Proceedings of the 2022 International Conference on Machine Learning and Cybernetics (ICMLC), Toyama, Japan, 9–11 September 2022.