





Automatic Medical Face Mask Recognition for COVID-19 Mitigation: Utilizing YOLO V5 Object Detection

Christine Dewi¹, Henoch Juli Christanto^{2*}

¹ Department of Information Technology, Satya Wacana Christian University, 52-60 Diponegoro Rd, Salatiga 50711, Indonesia

² Department of Information System, Atma Jaya Catholic University of Indonesia, Jakarta 12930, Indonesia

Corresponding Author Email: henoch.christanto@atmajaya.ac.id

<https://doi.org/10.18280/ria.370312>

ABSTRACT

Received: 1 May 2023

Accepted: 31 May 2023

Keywords:

face mask recognition, YOLO V3, YOLO V5, convolutional neural network, object detection

The ongoing COVID-19 pandemic has significantly affected global public health, necessitating protective measures such as wearing face masks to reduce the spread of the disease. Recent advances in deep learning-based object detection have shown promise in accurately recognizing objects within images and videos. In this study, the state-of-the-art You Only Look Once (YOLO) V5 object detection model was employed to classify individuals based on their mask-wearing status into three categories: none, poor, and adequate. YOLO V5 is known for its high efficiency and precision in object recognition tasks. Two datasets, the Face Mask Dataset (FMD) and the Medical Mask Dataset (MMD), were combined for simultaneous evaluation. The performance of the models was assessed based on crucial metrics such as Giga-Floating Point Operations (GFLOPS), workspace area, detection time, and mean average precision (mAP). Results indicated that the YOLO V5m model achieved the highest mAP (97.2%) for the "adequate" class, demonstrating its effectiveness in detecting proper mask usage for COVID-19 mitigation.

1. INTRODUCTION

Since the identification of the first case of coronavirus disease 2019 (COVID-19), the outbreak has rapidly escalated, leading to a global pandemic in 2020. On January 30, 2020, the World Health Organization (WHO) declared the situation a Public Health Emergency of International Concern (PHEIC), and the pandemic status was confirmed on March 11, 2020 [1]. The COVID-19 pandemic has posed a substantial challenge to the world, necessitating the deployment of various strategies to combat the virus, including the application of artificial intelligence (AI).

AI techniques have been employed in numerous capacities to aid in the fight against the virus, such as detecting individuals wearing face masks [2], identifying COVID-19 patients [3], and enhancing lesion segmentation in chest computed tomography (CT) scans of COVID-19 patients [4]. The integration of AI in these domains underscores its potential for mitigating the impact of the pandemic and facilitating informed decision-making processes.

Besides, WHO has provided guidance on the usage of face masks, and the studies [5, 6] have all supported the idea that wearing masks can prevent the spread of possible pathogens during a pandemic. One line of evidence suggests that wearing masks in public could be interpreted as a sign of social unity among those responding to the global pandemic. In order to combat COVID-19, Javid et al. suggest that we adopt a united strategy of wearing masks in public as our primary form of defense [7]. The face masks can be used twice, three times, or even four or five times provided they are kept in an atmosphere that is relatively safe and are well maintained, which is normal and usual, especially under the scenario when there is a shortage of face masks [8, 9].

Automated face mask detection plays a crucial role in enforcing public health measures and mitigating the spread of COVID-19. Here are some reasons why it is important: (1) Compliance with mask mandates: Wearing masks is a fundamental preventive measure recommended by health authorities to reduce the transmission of COVID-19. Automated face mask detection systems help enforce mask mandates in public spaces by identifying individuals who are not wearing masks or are wearing them incorrectly. By quickly identifying non-compliance, these systems can facilitate timely intervention and encourage adherence to mask-wearing guidelines. (2) Real-time monitoring: Automated face mask detection allows for real-time monitoring of mask usage in various settings, such as airports, public transportation, workplaces, and retail stores. This technology can provide continuous surveillance and alert authorities or personnel when individuals are not wearing masks. By identifying and addressing non-compliance promptly, potential outbreaks can be prevented or minimized [10]. (3) Efficient and consistent enforcement: Human monitoring of mask compliance can be challenging, especially in crowded areas or for extended periods. Automated systems provide a consistent and reliable means of enforcing mask-wearing policies without the need for constant human intervention. This enables authorities to allocate their resources more effectively and focus on other critical tasks related to public health and safety. (4) Public awareness and education: Face mask detection systems can serve as educational tools by raising awareness about the importance of mask-wearing. When people see the system in action, it reminds them to follow public health guidelines and encourages them to wear masks correctly. Over time, this increased visibility can help normalize mask usage and improve overall compliance rates [11]. (5) Data collection and

analysis: Automated face mask detection systems can generate valuable data on mask compliance rates, hotspots of non-compliance, and trends over time. This data can be analysed to identify patterns and make informed decisions regarding public health interventions. By understanding where and when mask usage is low, authorities can implement targeted educational campaigns, allocate resources, and adjust their strategies accordingly [12].

Furthermore, urgent research into the duration of protection provided by face masks [13], measures to extend the life of disposable masks, and the development of reusable masks should be encouraged. To combat and win the battle against the COVID-19 pandemic, the government must provide instructions and surveillance to individuals in public places, particularly in highly populated areas, according to the WHO. A component of this is making sure that the laws regarding face masks are followed. For instance, the combination of artificial intelligence models and surveillance technologies might be useful in this scenario [14, 15].

Artificial Intelligence (AI) has been applied in various areas, including mask detection, patient identification, and lesion segmentation, to improve efficiency and accuracy in healthcare settings. Some examples of AI applications are as follows: (1) Mask detection: AI-based computer vision algorithms can analyse images or video streams to detect whether individuals are wearing masks correctly or not. By using techniques like object detection and facial recognition, AI models can identify faces and determine if a mask is present and properly worn. This technology has been employed in public spaces, airports, hospitals, and other settings to enforce mask mandates and ensure compliance with public health guidelines [16, 17]. (2) Patient identification: AI has been used to improve patient identification processes in healthcare facilities. By analysing unique biometric identifiers such as facial features, fingerprints, or iris patterns, AI systems can accurately match patients to their electronic health records (EHRs) and prevent identification errors. This helps reduce medical errors, streamline administrative workflows, and enhance patient safety [18, 19]. (3) Lesion segmentation: In medical imaging, AI algorithms have been developed to automatically segment and analyse lesions in various modalities such as MRI, CT scans, and dermatological images [20]. These algorithms use deep learning techniques to identify and delineate the boundaries of lesions, assisting radiologists and dermatologists in diagnosing and monitoring conditions such as tumors, skin cancers, or abnormalities in organs. AI-based lesion segmentation can save time, improve accuracy, and aid in early detection and treatment planning [21]. (4) Medical image analysis: AI has revolutionized medical image analysis by enabling automated interpretation and diagnosis. Deep learning models trained on vast amounts of medical image data can detect and classify abnormalities, assist in radiological diagnoses, and provide quantitative measurements. AI algorithms have been applied to a wide range of medical imaging techniques, including X-rays, mammograms, ultrasounds, and pathology slides, improving efficiency and aiding clinicians in making more accurate diagnoses [22, 23].

In this article, a mask face detection model was developed with the assistance of deep transfer learning, and it is presented here in this study. Because the suggested model can differentiate between people who are and are not wearing masks, it is possible that this model might be combined with security cameras to prevent the transmission of COVID-19

and, as a result, prevent the transmission of COVID-19. The major contributions of this paper are as follows. First, a unique deep learning detection model that can automatically identify and localize facial medical masks on images has been created and presented. Second, evaluation of the advantages and disadvantages of using YOLO V5m and YOLO V5s to identify medical face masks were both included in this study. Next, we analyse our proposed method to the combination FMD and MMD dataset.

The organizational framework of this research is as follows: Section 2 contains the relevant work. Our recommended strategies are outlined in Section 3. In Section 4, we discuss experimental results and how they work. In the fifth and final part of this paper, conclusions are drawn and suggestions for additional research are offered.

2. RELATED WORK

2.1 Medical face mask recognition

Over the past few years, deep learning recognition has made significant breakthroughs in the majority of object recognition algorithms [24]. Identifying objects is easy for individuals, but it is incredibly challenging for computers to differentiate between two things that are virtually indistinguishable from one another in terms of their look and their functions. When people are obliged to wear face masks, a substantial amount of attention is typically focused on the formation of their faces as well as the identification of their genuine identities. This is the case in most situations in which people are required to wear face masks. In the study [25], the investigators are looking for those who are not using face masks in order to assist in the prevention and reduction of the transmission of the COVID-19 virus as well as other infections. A comparison of masked and non-masked face recognition datasets is provided in the study [26] along with a presentation of Principal Component Analysis (PCA). As part of their research, they have found statistical procedures that have the potential to be utilized in techniques for maskless face identification as well as masked face recognition. PCA is a frequently used statistical analysis technique that is more effective and efficient than other methods.

Researcher [27] are concentrating on the unmasking of a masked face since it is a really thoughtful young with significant practical implications. In their study, they employed a GAN-based network [16, 17] with two discriminators. Initially, one discriminator assisted in learning the overall structure of the face, and then another discriminator was inserted to concentrate learning on the deep missing region. GAN stands for Generative Adversarial Network. GAN-based networks are a type of neural network architecture that consists of two main components: a generator network and a discriminator network. The concept of GANs was introduced by Ian Goodfellow and his colleagues in 2014 [28, 29].

The generator network in a GAN generates new data samples, such as images or text, by learning from existing data. It takes random input, often called noise, and transforms it into synthetic data that resembles the training data. The goal of the generator is to create samples that are realistic and indistinguishable from the real data.

The discriminator network, on the other hand, acts as a binary classifier. It receives both real data samples from the training set and generated samples from the generator. The

discriminator's objective is to distinguish between real and fake data. It learns to assign high probabilities to real samples and low probabilities to generated samples [30].

Researchers [31] presented a method for identifying a person with a masked face in an image or video stream. These two technologies were employed in this study. The project is being carried out in two parts. In the first phase, a deep learning model is trained, and then, in the second phase, the mask detector is applied to an image or video stream that is being streamed live. To do real-time face detection from a webcam live feed, OpenCV is used. On a dataset, the COVID-19 face mask detector was constructed with the help of computer vision and Python.

Another research [32] has published one work in which the proposed system focuses on recognizing masks, and faces are represented using the more advanced YOLOV3 architecture. YOLO, which stands for "You Only Look Once," uses a learning process known as the Convolution Neural Network (CNN) [33]. A Convolutional Neural Network (CNN) is a type of deep learning neural network that is particularly effective in processing and analyzing data with a grid-like structure, such as images and videos. CNNs have revolutionized computer vision tasks by demonstrating superior performance in tasks like image classification, object detection, and image segmentation. The key component of a CNN is the convolutional layer. This layer applies a series of filters or kernels to the input data, extracting features by performing convolution operations. Each filter detects specific patterns or features in the input, such as edges, textures, or shapes. The convolution operation involves sliding the filter across the input, computing dot products at each position, and producing a feature map that highlights the presence of that specific feature. YOLO is capable of detecting and finding all kinds of images and has a relationship with CNN that has been built through hidden layers, research, and easy search algorithms [34, 35]. After combining the results to provide action-level predictions, the execution phase begins with incorporating thirty different images from the data set into the model. In addition to providing great image results, it is also capable of detecting objects. Using this model, we can see how the model performs with masked and unmasked layers, and how fast the frame rate is when incorporated in a video stream [36, 37].

RetinaFaceMask is the name of the face mask detector that was proposed by Mingjie Jiang and his colleagues [38]. Their approach involved the utilization of an innovative object-removal technique to get rid of forecasts that had a low level of confidence. They were able to get results that were 5.9 percent and 11.0 percent more accurate recall, as well as 1.5 percent and 2.3 percent greater precision, compared to the most recent state-of-the-art results for face mask and face detection, respectively. On the other hand, they also investigated how well the suggested strategy performed on lightweight neural networks like MobileNet [39].

2.2 YOLO V5

YOLO V5 is an evolution of the YOLO algorithm and introduces architectural changes and improvements over previous versions. It is not specifically designed for medical face mask recognition, but it can be utilized for this task with appropriate training on a labeled dataset of masked and unmasked faces.

To adapt YOLOv5 for medical face mask recognition, the following steps can be taken: (1) Dataset collection and

annotation: Collect a diverse dataset of images containing faces with and without masks. Annotate the images with bounding boxes around the faces and label them accordingly as masked or unmasked. (2) Model configuration: Modify the YOLOv5 architecture to suit the specific requirements of medical face mask recognition. Adjust the input size, number of classes (masked and unmasked), and anchor box sizes according to your dataset. (3) Training: Train the YOLOv5 model on the annotated dataset using the modified configuration. This involves optimizing the model's weights to learn to accurately detect and classify faces with and without masks. (4) Evaluation and fine-tuning: Evaluate the trained model's performance on a validation dataset to measure its accuracy and adjust the hyperparameters if needed. Fine-tuning can be performed to improve the model's performance further [40]. (5) Testing and deployment: Test the trained YOLOv5 model on a separate test dataset to assess its real-world performance. Once satisfied with the results, deploy the model to perform medical face mask recognition tasks. It's important to note that developing a robust and accurate medical face mask recognition system requires a diverse and representative dataset, careful model configuration, and thorough training and evaluation. Additionally, considering ethical and privacy aspects is crucial when working with medical data or deploying such systems in healthcare settings [41].

There are five unique designs for the YOLO V5's architecture, and they are the YOLO V5s, YOLO V5m, YOLO V5n, and YOLO V5l designs [42, 43]. The primary difference between the two is the number of scattered feature extraction modules and convolution kernels across the network at various nodes. Figure 1 provides a schematic representation of the internal network that YOLO V5 possesses and may be found in this paper. The YOLOV5 design makes use of a wide variety of technologies, such as autonomous learning bounding box anchoring, mosaic data enhancement, and cross-stage partial networking, amongst others. An initial image with dimensions of 608x608x3 is inserted into the Focus structure, as illustrated in Figure 1. This is done with reference to the YOLO V5 structure, which serves as an example. After this step, the image is transformed into a feature map that has dimensions of 304 x 304 x 12, then continues with the kernel convolution operation 32, resulting in a final feature map having dimensions of 304 x 304 x 32. This architecture makes advantage of some of the most powerful algorithm optimization methods that have been created for convolutional neural networks in the most recent handful of years. It utilizes the YOLO detection architecture as its foundation. In our experiment, we only employ YOLO V5s and YOLO V5m and compare with YOLO V3 and YOLO V3 SPP [44, 45].

YOLO V5 has four main components: input, backbone, neck, and output [46]. The major responsibility of the Backbone Model is to identify significant pieces for analysis from inside the input image. Cross Stage Partial Networks (CSP) and Spatial Pyramid Pooling (SPP) [47] are the primary building blocks that YOLO V5 uses when it comes to extracting rich and important characteristics from input photographs. This is accomplished with the help of YOLO V5. When it comes to the accurate generalization of a model for object scaling, it is vital to correctly identify the same item in many sizes and scales. SPP is helpful in this regard because it can recognize the same thing in different sizes and scales. The feature pyramid architectures of Feature Pyramid Network (FPN) [48] and Path Aggregation Network (PANet) [49] are

utilized in the construction of the neck network. The FPN structure has significant semantic features that are scattered across its entirety, starting at the top feature maps and working their way down to the lower feature maps. These features begin their journey at the top feature maps. During this time, it is the role of the PAN structure to ensure that trustworthy

localization features are transmitted from lower feature maps to higher feature maps. Feature maps have a hierarchical structure. PANet is utilized by YOLO V5 in the capacity of a neck, which makes the development of a feature pyramid possible.

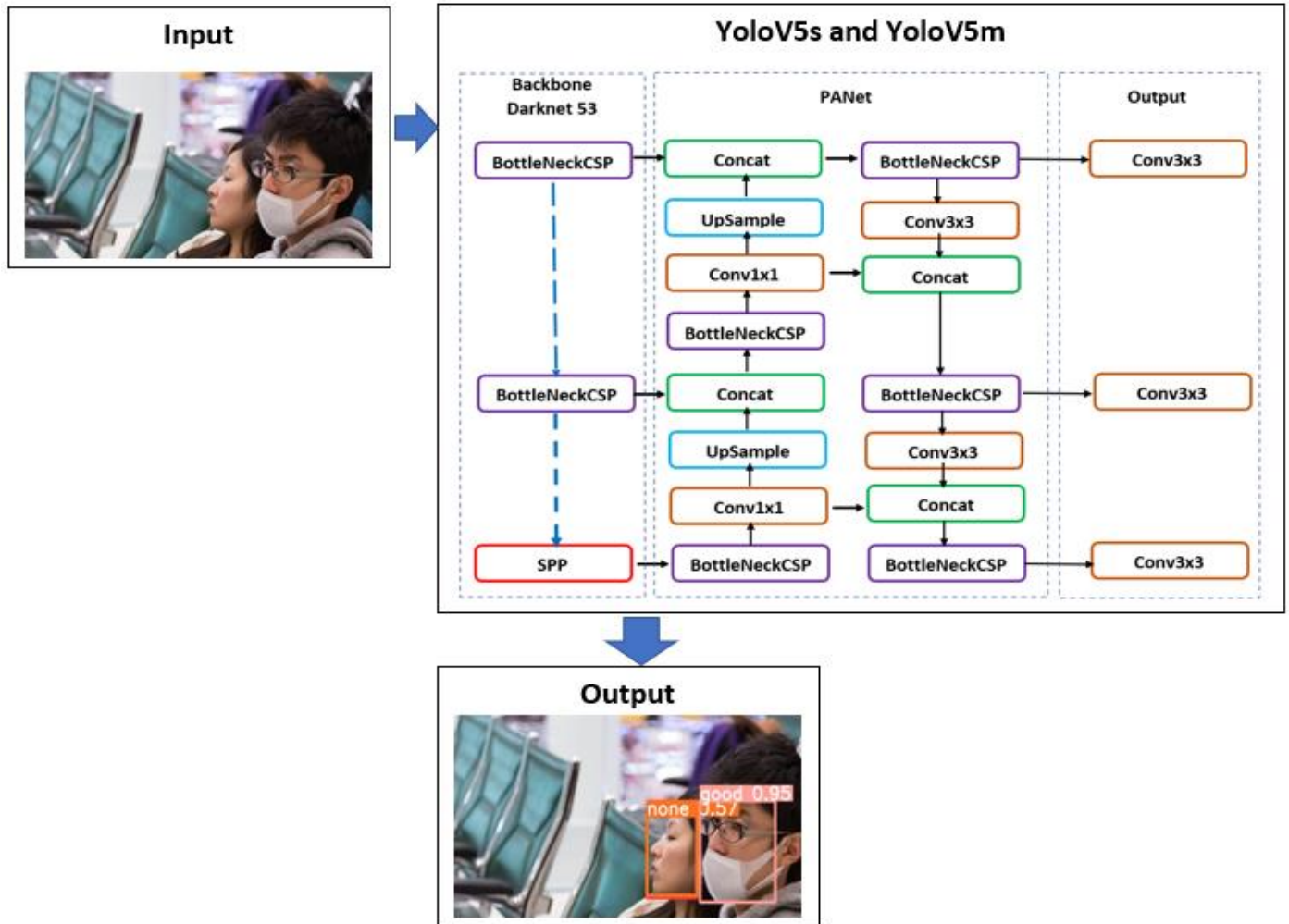


Figure 1. YOLOV5 architecture

Important advances were made in YOLO version 5, including the following: Because YOLO V5 is based on PyTorch and not a derivative of the original Darknet code, it differs from all previous editions of the software. YOLO V5, in the same way as YOLO V4, has a CSP backbone and a PANET neck [49]. The most significant improvements include improved mosaic data and automatic learning of bounding box anchors.

Yolo loss function based on Eq. (1) [50].

$$\begin{aligned}
 & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y - \hat{y}_i)^2] + \\
 & \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + \left(\sqrt{h_i} - \right. \right. \\
 & \left. \left. \sqrt{\hat{h}_i} \right)^2 \right] + \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \\
 & \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \\
 & \sum_{i=0}^{s^2} \mathbb{I}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned} \quad (1)$$

3. METHODOLOGY

3.1 The combination of Face Mask Dataset (FMD) and Medical Masks Dataset (MMD)

The studies described in this work were carried out utilizing two datasets of medical face masks that were made available to the public. To begin, the Face Mask Dataset (FMD) [51] is the masked face dataset that is freely accessible to the public. The FMD dataset is comprised of 853 images, all of which are in the PASCALVOC format and are included in this collection. In addition, FMD datasets are separated into three categories: (1) with mask, (2) without mask and (3) mask worn incorrectly. These images contain a total of 4072 faces, 3232 of which are identified as with masks, 717 as without masks, and 123 as mask worn incorrectly. Figure 2(a) illustrates various FMD example images. The next resource is the Medical Masks Dataset (MMD), which may be found on Kaggle [52].

In addition, there are a total of 9067 labeled faces in the MMD dataset, including 6758 faces with masks, 2085 faces without masks, and 224 faces with improperly worn masks;

these labels are useful for researching issues related to face-mask recognition as shown in Table 1. Examples of images that can be created with MMD are displayed in Figure 2(b). The YOLO format requires that a text file with the same name as the associated image file be created and saved in the same directory as the image file, but with the extension changed

to.txt. Include the object number as well as the object coordinates on this image within the txt file. Place the following information for each item in a new line: <object-class><x><y><width><height>. Before beginning the training on the dataset, we convert the labels from the PASCAL VOC format to the YOLO format.



Figure 2. Sample image of FMD and MMD dataset

Table 1. FMD and MMD dataset information

Dataset	Type	Mask Type	Image	Faces	Available Labels		
					Mask	No Mask	Incorrect Worn
FMD [51]	Image Dataset	Real-world	853	4072	3232	717	123
MMD [52]	Image Dataset	Real-world	6024	9067	6758	2085	224

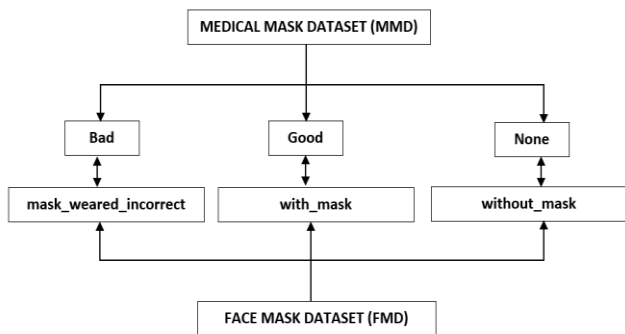


Figure 3. The combination of MMD and FMD Dataset

In our experiment, MMD and FMD were merged into a single data set to produce a distinctive result. After removing images of poor quality and duplicates from the source dataset, a total of 6877 photographs were merged from the one that was provided. This was achieved by combining the photographs. The combination of MMD and FMD is depicted in Figure 3, which may be found in our publications. The MMD dataset is comprised of three classes, referred to as terrible, good, and none respectively. However, the FMD dataset distinguishes between three groups: "mask wear incorrect," "with mask," and "without mask." Our investigation led us to classify people into three groups: those who wore their masks improperly ("bad"), those who did not ("good"), and those who did not ("none").

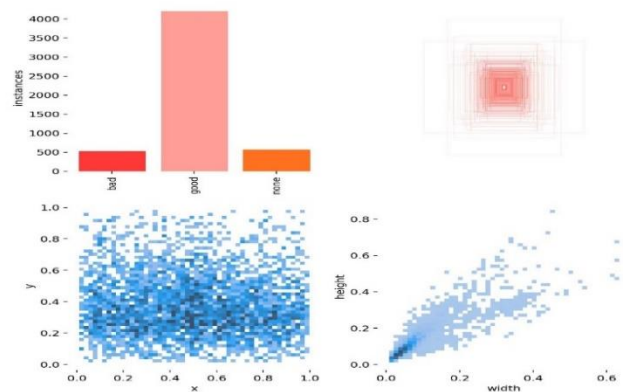


Figure 4. Labels of the FMD and MMD dataset

The labels of the MMD and FMD datasets are shown in Figure 4 and these datasets have three classes: bad, good, and none. The bad class consist of around 500 instances, class good more than 4000 instances, and class none with 400 instances. Furthermore, the width of the dataset ranges from 0.0 to 0.6, and the height ranges from 0.0 to 0.8.

3.2 Training result

The training environment for the face mask model featured an AMD Ryzen 7 3700X Central Processing Unit (CPU) with

an 8-core processor, an Nvidia GTX2070 Super GPU accelerator, and 32GB of DDR4-3200 memory. An AMD Ryzen 7 3700X computer served as a home for each of these individual components. During the training phase, we divided our dataset into two parts: 70 percent for training and 30 percent for testing. Whenever a training batch is processed, YOLO V5 sends the accumulated training data to a data loader. The data loader then augments the data online, which is displayed to the user. The data loader is responsible for performing the following types of augmentations: scaling adjustments, updates to the color space, and mosaic augmentation. The mosaic data enhancement approach is by far the most original, as it mixes four pictures into four tiles with a random aspect ratio. This makes the mosaic method the clear winner. In contrast to Darknet's use of a.cfg files, YOLO V5 creates its model configurations using the a.yaml file format. The most significant distinction between these two formats is that the a.yaml file is a simplified version that only specifies the various layers in the network and then multiplies

those by the total number of layers in the block [53, 54].

Figure 5 exhibits the training and testing graph with (a) YOLO V3 and (b) YOLO V5s. The training for each of the models used in the experiment consisted of a total of 40 epochs. The training for the YOLO V3 SPP was finished in a total of 14,515 hours, the YOLO V3 was accomplished in a total of 14,473 hours, the YOLO V5s was done in a total of 2,969 hours, and the YOLO V5m was completed in a total of 6,569 hours. The last step in the training process is termed fine-tuning, and participation in it is totally voluntary. The learning rate is a hyperparameter that indicates how significant of a modification should be made to the model in order to account for the projected error each time the weights of the model are updated. In this phase, we will disassemble the entire model that we obtained in the previous stage, and then we will retrain it on our data while setting the learning rate to a very low value. By gradually adjusting previously trained features to take into account newly acquired data, it has the potential to bring about significant gains in performance.

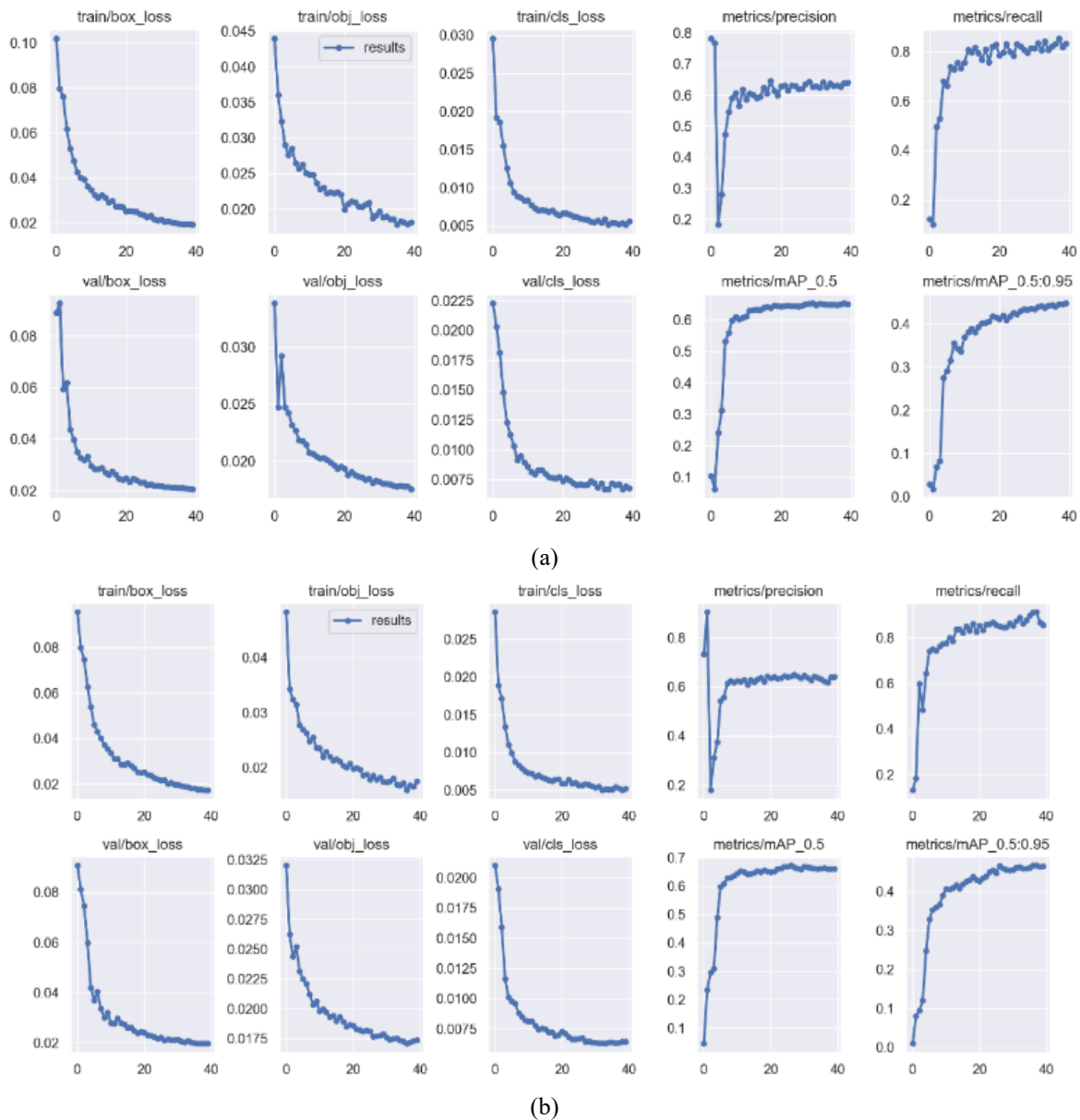


Figure 5. Training and testing graph with the combination of FMD and MMD dataset. (a) YOLO V3 and (b) YOLO V5s

Table 2. Training performance results for all models with FMD and MMD dataset

Model	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
YOLO V3 SPP	all	507	2661	0.643	0.85	0.652	0.465
	bad	507	260	0.517	0.835	0.534	0.352
	good	507	2123	0.927	0.956	0.961	0.701
	none	507	278	0.484	0.76	0.519	0.343
YOLO V3	all	507	2661	0.643	0.872	0.651	0.479
	bad	507	260	0.503	0.861	0.531	0.357
	good	507	2123	0.941	0.953	0.962	0.723
	none	507	278	0.484	0.801	0.517	0.358
YOLO V5s	all	507	2661	0.639	0.832	0.662	0.448
	bad	507	260	0.508	0.815	0.492	0.317
	good	507	2123	0.933	0.945	0.963	0.697
	none	507	278	0.476	0.736	0.496	0.329
YOLO V5m	all	507	2661	0.639	0.832	0.672	0.448
	bad	507	260	0.508	0.815	0.492	0.317
	good	507	2123	0.933	0.945	0.964	0.697
	none	507	278	0.476	0.736	0.496	0.329

The hyperparameters-configurations file is where you will make any necessary adjustments to the learning rate settings to meet your requirements. Our work makes advantage of the hyperparameters described in the built-in `hyp.finetune.yaml` file so that the tutorial can better demonstrate their application. This hyperparameter has a significantly reduced learning rate compared to the one that is defaulted. The initial value of the weight will be determined by the value that was saved in the stage before this one. In addition, the results of our training performance with FMD and MMD datasets are described in Table 2, which may be found here. According to the findings in Table 2, the YOLO V5m has the highest average mAP, which comes in at 67.2%, while the YOLO V5s shows 66.2% mAP.

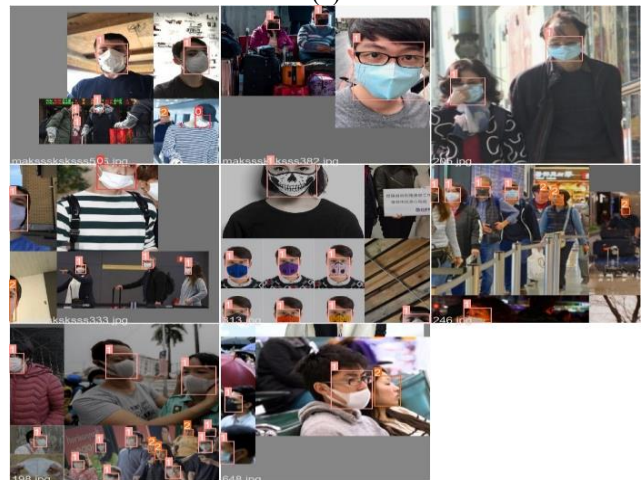
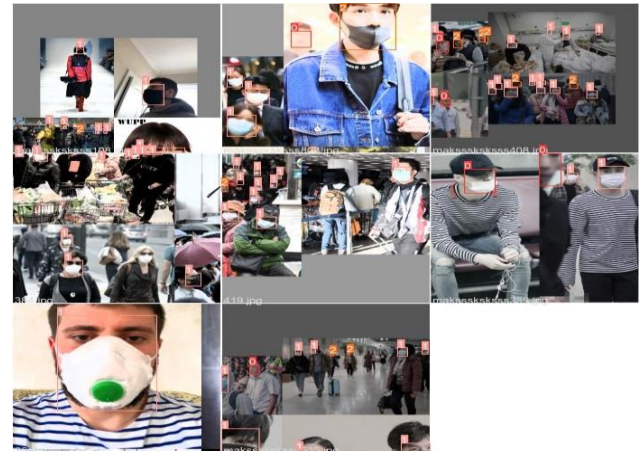
Figure 6 depicts the training technique that was taken for both batch 0 and batch 1, respectively. Anchor boxes were created by YOLO V5 using a genetic algorithm as their basis. An automatic anchoring process is what they name this method, which recalculates the anchor boxes to better fit the data if the default ones are insufficient. The K-means method is paired with this information to generate a k-means evolved anchor box. This is one of the reasons why YOLO V5 performs so well even when applied to a wide variety of datasets.

The process of generating anchor boxes in YOLOv5 are as follows: (1) Dataset preparation: The first step is to prepare a labeled dataset containing bounding box annotations for the objects of interest. The dataset should cover a diverse range of object sizes and aspect ratios. (2) Extracting anchor box dimensions: From the dataset, the width and height of the ground truth bounding boxes are extracted. These dimensions are usually normalized by the image width and height to ensure consistency across different images. (3) Running k-means clustering: The k-means clustering algorithm is applied to the extracted bounding box dimensions. The aim is to group similar-sized objects together and determine representative anchor box dimensions for each group. The number of clusters (k) corresponds to the desired number of anchor boxes. (3) Computing anchor box dimensions: After clustering, the centroid of each cluster represents an anchor box. The centroid's width and height values are converted back to the original scale (pixel values) by multiplying them with the image width and height. (4) Finalizing anchor box sizes: Depending on the specific implementation of YOLOv5, additional adjustments might be made to the anchor box sizes to better match the distribution of objects in the dataset or to

achieve specific performance goals [55].

Moreover, *mAP* is calculated employing Intersection over Union (IoU). Specifies the degree of overlap between the expected and ground truth bounding boxes. A value of 0 for the IoU indicates that there is no overlap between the boxes. An IoU of 1 indicates that the union of the boxes is equal to their overlap, suggesting that they overlap entirely. IoU describes in the Eq. (2) [56].

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}} \quad (2)$$

**Figure 6.** Training Process (a) batch 0 and (b) batch 1

Additionally, the output samples have the possibility to be separated into three different categories according to their characteristics. A result is a true positive (TP) when the model correctly predicts the presence of the positive class. A result is a true negative if it indicates that the model successfully predicted the negative class. A result is said to have a false positive (FP) classification when the model incorrectly predicts the presence of the positive class. A false negative (often abbreviated as FN) is an outcome that occurs when the model incorrectly forecasts the negative class. The number of samples for which a positive result was wrongly assigned is referred to as the "true negative" (TN) count. Precision and recall are represented by [57, 58] in Eqs. (3)-(4). Further, F1 [59] is shown in Eq. (5) [50].

$$Precision (P) = \frac{TP}{TP+FP} \tag{3}$$

$$Recall (R) = \frac{TP}{TP+FN} \tag{4}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision+ Recall} \tag{5}$$

Figure 7 provides detailed information regarding the detection performance and amount of complexity of several different models, which are broken down into categories such as layers, parameters, and GFLOPS respectively. After that, the YOLO V3 SPP has 269 layers, and the YOLO V3 load has the same 261 layers. The YOLO V5s and YOLO V5m contain the most layers, around 290 layers, according to our experiment. YOLO V3 SPP loads the most 62557288 parameters, followed by YOLO V3 with 61508200 parameters, YOLO V5s and YOLOV5m contain the same parameters 20861016. The YOLO V5s and YOLO V5m require a workspace size of 41.2 MB. In addition, YOLO V3 SPP produces a total of 155.6 GFLOPS, allocates an additional 125.5 MB of workspace space, and YOLO V3 provides a big 123.4 MB workspace space while producing a total of 154.7 GFLOPS. Together, the YOLO V5s and YOLO V5m are only capable of producing a paltry 48 GFLOPS.

When compared to competing object detection frameworks,

YOLO V5's ease of use for developers integrating computer vision technology into applications is outstanding. The following is a list of some of the facilities offered by YOLO V5: (1) Simple Setup – YOLO V5 only requires a torch and some lightweight python libraries to be installed on your computer to work. (2) Fast Learning: The YOLO V5 model learns very quickly, which allows us to reduce the amount of money spent on experiments as we build our model. (3) Ports for inference that work, and we can use YOLO V5 to infer individual photos, image sets, video feeds, or webcam ports. (4) Easy to Traverse Layout The file folder layout is easy to understand and navigate as you develop. (5) Simple Portability to Mobile Devices - We were able to quickly port YOLO V5 from *PyTorch* weights to *ONNX* weights to *CoreML* to *IOS*.

The smaller YOLO V5 model has approximately 2.5 times faster run time while achieving higher performance in recognizing small objects. Also, very little to no overlapping squares due to this result, making it much cleaner. *Ultralytics* has done an excellent job on their open-source YOLO V5 model, which makes it easy to inference and train the model.

4. RESULT AND DISCUSSIONS

The effectiveness of three types of facemask identification is summarized in Table 3. We employ YOLO V3, YOLO V3 SPP, YOLO V5s and YOLO V5m in this experiment.

The results of our evaluation of the YOLO V5 series are detailed in Table 3. Our model is now ready to move on to the inference phase now that it has successfully completed the training phase and achieved the desired results. The final forecast is an ensemble of all the augmented images (obtained by flipping the images horizontally and choosing one of three resolutions). We implement mosaic data augmentation, and the steps are as follows: (1). Resize the images to be the same dimensions. (2). Sample from the Beta distribution to get the λ value. (3). Multiply all values in image 1 by λ . (4). Multiply all values in image 2 by $1-\lambda$. (5). Add the two images together to get the final image. (6). Combine the annotations to get the final annotations for the image.

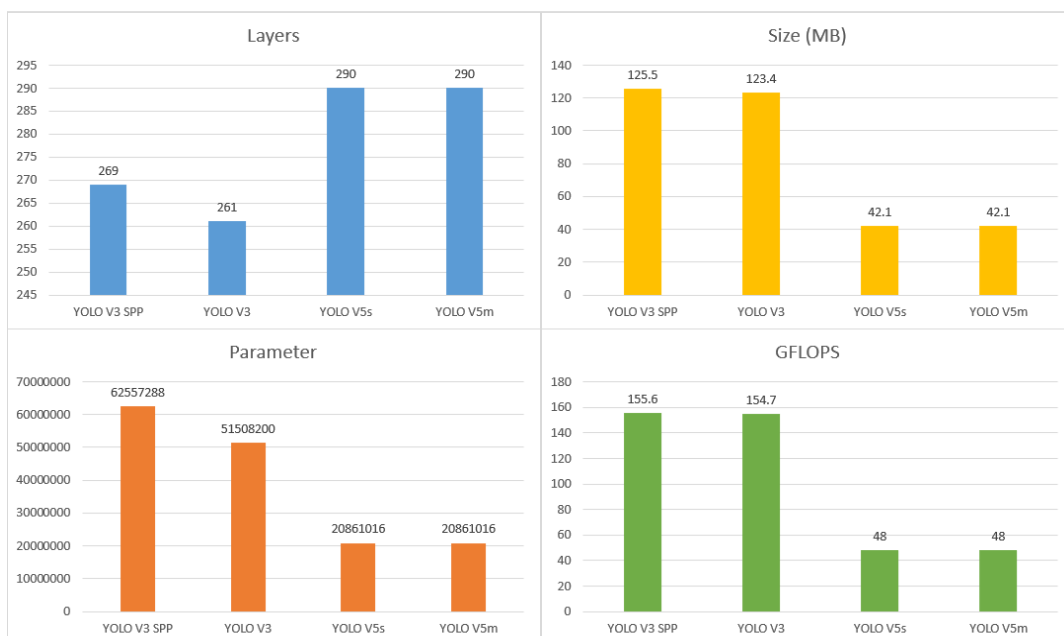


Figure 7. The comparison of layers, size, parameter and GFLOPS

Table 3. Testing accuracy results performance for all models with FMD and MMD dataset

Model	Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95
YOLO V3 SPP	all	507	2661	0.624	0.882	0.652	0.453
	bad	507	260	0.495	0.842	0.501	0.322
	good	507	2123	0.903	0.961	0.96	0.702
	none	507	278	0.475	0.842	0.526	0.334
YOLO V3	all	507	2661	0.647	0.87	0.651	0.471
	bad	507	260	0.522	0.819	0.516	0.333
	good	507	2123	0.936	0.958	0.961	0.724
	none	507	278	0.484	0.833	0.525	0.357
YOLO V5s	all	507	2661	0.615	0.837	0.662	0.436
	bad	507	260	0.475	0.777	0.48	0.296
	good	507	2123	0.932	0.958	0.97	0.728
	none	507	278	0.471	0.791	0.522	0.336
YOLO V5m	all	507	2661	0.626	0.886	0.671	0.46
	bad	507	260	0.481	0.858	0.523	0.336
	good	507	2123	0.931	0.957	0.972	0.713
	none	507	278	0.465	0.845	0.509	0.332



Figure 8. Validation test batch 2 with YOLO V5m



Figure 9. Recognition result class none for all models

Inclusion of test-time augmentations (TTA), which comes after inference, enables us to further improve the accuracy of the predictions. If we want to maintain a high frames-per-second (FPS) rate, we will have to forego the use of the TTA because the inference it generates is two to three times as long as it would be otherwise. TTA is a technique used to improve the accuracy of object detection models by applying data augmentation during the inference stage. It involves augmenting test images multiple times with various transformations and aggregating the predictions from these augmented images. While TTA can enhance the model's

accuracy, it comes with trade-offs in terms of inference time and FPS. When applying TTA, the inference time increases because the model needs to make predictions for each augmented version of the test image. The additional computations for each augmentation can significantly slow down the inference process, resulting in a lower FPS rate. The relationship between TTA and FPS can be inversely proportional: as the number of augmentations increases, the FPS rate decreases. The trade-offs between using TTA for improved accuracy and maintaining a high FPS rate depend on the specific application and its requirements.



Figure 10. Face mask recognition result with YOLOV5m

Based on our experiment result on Table 2, YOLO V5m achieve the highest average *mAP* of 67.1% for all classes. Followed by YOLO V5s with a *mAP* of 66.2%, YOLO V3 SPP with a *mAP* of 65.2%, and YOLO V3 with a *mAP* of 65.1%. Class good achieves the highest *mAP* for all models, ranging from 96% to 97.2%. When deep learning is being carried out, some parameters, known as hyperparameters, are set before the formal training begins. It is possible that the performance of the model can be improved by making use of the relevant hyperparameters. There is a total of 23 hyperparameters in the YOLO V5 algorithm, the vast majority of which are involved in customizing aspects like learning velocity, loss function, and data improvement settings.

The validation procedure for YOLO V5m batch 2 is shown in Figure 8. During the training phase of YOLO V5, four separate images were mixed to produce one larger image. During the splicing phase, each of the four independent images goes through a random processing step. This causes each image to have a distinct difference in size and configuration. Nevertheless, the recognition result class “Good” and “None” for all models shown in Figure 9. Most of the model in the experiment can recognize the class “Good” and “None” very well. In addition, Figure 10 illustrates the various recognition result with YOLO V5m.

5. CONCLUSIONS

The purpose of this research is to present a clear and concise summary of CNN-based object recognition methods, with an emphasis on the SPP algorithms of YOLO V5s, YOLO V5m, YOLO V3, and YOLO V3. During our experimental research, we put several current object detectors through their paces and evaluated how well they perform. Some of the detectors we saw included, for example, those meant to recognize facemask. The scoring criteria included measurements for a variety of important features, including mean acquisition time (*mAP*), detection time, *IoU*, and number of GFLOPS.

Based on our experiment result, YOLO V5m achieved the highest average *mAP* of 67.1% for all classes during training and testing stage. Next, Class “Good” achieves the highest *mAP* for all models, ranging from 96% to 97.2%. We focus on Class “Good” that means our algorithm can detect people wearing masks precisely. Facemask identification technology has significant real-world implications for enforcing mask-wearing policies in public spaces and improving public health during a pandemic. It is important to implement facemask identification technology in a manner that respects privacy and addresses ethical considerations. Transparency, consent, and data protection measures should be in place to ensure the responsible use of this technology while balancing public

health needs and individual rights. Additionally, combining facemask identification technology with other preventive measures, such as vaccination, testing, and social distancing, is essential for comprehensive public health strategies during a pandemic.

The dataset that uses in our research is imbalanced, so it needs to add more dataset in the future to solve the imbalanced data problem. Moreover, in the future we will enlarge the dataset, add not ideal image or noise, and combine it with explainable artificial intelligence (XAI).

REFERENCES

- [1] Wang, B., Zheng, J., Chen, C.P. (2021). A survey on masked facial detection methods and datasets for fighting against COVID-19. *IEEE Transactions on Artificial Intelligence*, 3(3): 323-343. <https://doi.org/10.1109/TAI.2021.3139058>
- [2] Loey, M., Manogaran, G., Taha, M.H.N., Khalifa, N.E.M. (2021). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement*, 167: 108288. <https://doi.org/10.1016/j.measurement.2020.108288>
- [3] Shaban, W.M., Rabie, A.H., Saleh, A.I., Abo-Elsoud, M.A. (2021). Detecting COVID-19 patients based on fuzzy inference engine and Deep Neural Network. *Applied Soft Computing*, 99: 106906. <https://doi.org/10.1016/j.asoc.2020.106906>
- [4] Mahmud, T., Rahman, M.A., Fattah, S.A., Kung, S.Y. (2021). CovSegNet: A multi encoder–decoder architecture for improved lesion segmentation of COVID-19 chest CT scans. *IEEE Transactions on Artificial Intelligence*, 2(3): 283-297. <https://doi.org/10.1109/TAI.2021.3064913>
- [5] Howard, J, Huang, A., Li, Z., Tufekci, Z., Zdimal, V., van der Westhuizen, H., von Delft, A., Price, A., Fridman, L., Tang, L., Tang, V., Watson, G.L., Bax, C.E., Shaikh, R., Questier, F., Hernandez, D., Chu, L.F., Ramirez, C.M., Rimoin, A.W. (2021). An evidence review of face masks against COVID-19. *Proceedings of the National Academy of Sciences*, 118(4): e2014564118. <https://doi.org/10.1073/pnas.2014564118>
- [6] Loey, M., Manogaran, G., Taha, M.H.N., Khalifa, N.E.M. (2021). Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustainable Cities and Society*, 65: 102600. <https://doi.org/10.1016/j.scs.2020.102600>
- [7] Javid, B., Weekes, M.P., Matheson, N.J. (2020). COVID-19: Should the public wear face masks?. *BMJ*, 369. <https://doi.org/10.1136/bmj.m1442>
- [8] Qin, B., Li, D. (2020). Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19. *Sensors*, 20(18): 5236. <https://doi.org/10.3390/s20185236>
- [9] Marini, M., Ansani, A., Paglieri, F., Caruana, F., Viola, M. (2021). The impact of facemasks on emotion recognition, trust attribution and re-identification. *Scientific Reports*, 11(1): 1-14. <https://doi.org/10.1038/s41598-021-84806-5>
- [10] Dewi, C., Chen, A.P.S., Christanto, H.J. (2023). YOLOv7 for face mask identification based on deep

- learning. In 2023 15th International Conference on Computer and Automation Engineering (ICCAE), Sydney, Australia, pp. 193-197. <https://doi.org/10.1109/ICCAE56788.2023.10111427>
- [11] Dewi, C., Chen, A.P.S., Christanto, H.J. (2023). Deep learning for highly accurate hand recognition based on Yolov7 model. *Big Data and Cognitive Computing*, 7(1): 53. <https://doi.org/10.3390/bdcc7010053>
- [12] Dewi, C., Chen, R.C., Zhuang, Y.C., Jiang, X., Yu, H. (2023). Recognizing road surface traffic signs based on YOLO models considering image flips. *Big Data and Cognitive Computing*, 7(1): 54. <https://doi.org/10.3390/bdcc7010054>
- [13] Singh, S., Ahuja, U., Kumar, M., Kumar, K., Sachdeva, M. (2021). Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimedia Tools and Applications*, 80: 19753-19768. <https://doi.org/10.1007/s11042-021-10711-8>
- [14] Lemke, M.K., Apostolopoulos, Y., Sönmez, S. (2020). Syndemic frameworks to understand the effects of COVID-19 on commercial driver stress, health, and safety. *Journal of Transport & Health*, 18: 100877. <https://doi.org/10.1016/j.jth.2020.100877>
- [15] Dewi, C., Chen, R.C., Liu, Y.T. (2022). Synthetic traffic sign image generation applying generative adversarial networks. *Vietnam Journal of Computer Science*, 9(3): 333-348. <https://doi.org/10.1142/S2196888822500191>
- [16] Himeur, Y., Al-Maadeed, S., Varlamis, I., Al-Maadeed, N., Abualsaud, K., Mohamed, A. (2023). Face mask detection in smart cities using deep and transfer learning: Lessons learned from the COVID-19 pandemic. *Systems*, 11(2): 107. <https://doi.org/10.3390/systems11020107>
- [17] Dewi, C., Tsai, B.J., Chen, R.C. (2022). Shapley additive explanations for text classification and sentiment analysis of internet movie database. In: Szczerbicki, E., Wojtkiewicz, K., Nguyen, S.V., Pietranik, M., Krótkiewicz, M. (eds) *Recent Challenges in Intelligent Information and Database Systems. ACIIDS 2022. Communications in Computer and Information Science*, vol. 1716. Springer, Singapore. https://doi.org/10.1007/978-981-19-8234-7_6
- [18] Suclupe, S., Kitchin, J., Sivalingam, R., McCulloch, P. (2022). Evaluating patient identification practices during intrahospital transfers: A human factors approach. *Journal of Patient Safety*, 19(2): 117-127. [10.1097/PTS.0000000000001074](https://doi.org/10.1097/PTS.0000000000001074)
- [19] Dewi, C., Chen, R.C., Yu, H., Jiang, X. (2023). XAI for Image Captioning using SHAP. *Journal of Information Science and Engineering*, 39(4): 711-724. [https://doi.org/10.6688/JISE.202307_39\(4\).0001](https://doi.org/10.6688/JISE.202307_39(4).0001)
- [20] Dewi, C., Chen, R.C. (2022). Complement naive bayes classifier for sentiment analysis of internet movie database. In: Nguyen, N.T., Tran, T.K., Tukayev, U., Hong, TP., Trawiński, B., Szczerbicki, E. (eds) *Intelligent Information and Database Systems. ACIIDS 2022. Lecture Notes in Computer Science*, vol. 13757. Springer, Cham. https://doi.org/10.1007/978-3-031-21743-2_7
- [21] Hasan, M.K., Ahamad, M.A., Yap, C.H., Yang, G. (2023). A survey, review, and future trends of skin lesion segmentation and classification. *Computers in Biology and Medicine*, 155: 106624. <https://doi.org/10.1016/j.compbiomed.2023.106624>
- [22] Hu, M., Zhang, J., Matkovic, L., Liu, T., Yang, X. (2023). Reinforcement learning in medical image analysis: Concepts, applications, challenges, and future directions. *Journal of Applied Clinical Medical Physics*, 24(2): e13898. <https://doi.org/10.1002/acm2.13898>
- [23] Dewi, C., Juli Christanto, H. (2022). Combination of deep cross-stage partial network and spatial pyramid pooling for automatic hand detection. *Big Data and Cognitive Computing*, 6(3): 85. <https://doi.org/10.3390/bdcc6030085>
- [24] Zhang, X., Wan, F., Liu, C., Ji, X., Ye, Q. (2021). Learning to match anchors for visual object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 3096-3109. <https://doi.org/10.1109/TPAMI.2021.3050494>
- [25] Said, Y. (2020). Pynq-YOLO-Net: An embedded quantized convolutional neural network for face mask detection in COVID-19 pandemic era. *International Journal of Advanced Computer Science and Applications*, 11(9): 100-106.
- [26] Ejaz, M.S., Islam, M.R., Sifatullah, M., Sarker, A. (2019). Implementation of principal component analysis on masked and non-masked face recognition. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, pp. 1-5. <https://doi.org/10.1109/ICASERT.2019.8934543>
- [27] Din, N.U., Javed, K., Bae, S., Yi, J. (2020). A novel GAN-based network for unmasking of masked face. *IEEE Access*, 8: 44276-44287. <https://doi.org/10.1109/ACCESS.2020.2977386>
- [28] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A and Bengio, Y (2014). Generative Adversarial Nets (NIPS version). *Advances in Neural Information Processing Systems*, 2672-2680. <https://arxiv.org/pdf/1406.2661.pdf>.
- [29] Dewi, C., Chen, R.C., Liu, Y.T., Yu, H. (2021). Various generative adversarial networks model for synthetic prohibitory sign image generation. *Applied Sciences*, 11(7): 2913. <https://doi.org/10.3390/app11072913>
- [30] Dewi, C., Chen, R.C., Liu, Y.T. (2021). Wasserstein generative adversarial networks for realistic traffic sign image generation. In: Nguyen, N.T., Chittayasothorn, S., Niyato, D., Trawiński, B. (eds) *Intelligent Information and Database Systems. ACIIDS 2021. Lecture Notes in Computer Science*, vol. 12672. Springer, Cham. https://doi.org/10.1007/978-3-030-73280-6_38
- [31] Suganthalakshmi, R., Hafeeza, A., Abinaya, P., Devi, A. G. (2021). COVID-19 facemask detection with deep learning and computer vision. *International Journal of Engineering Research & Technology (IJERT)*, 9(5): 73-75.
- [32] Bhuiyan, M.R., Khushbu, S.A., Islam, M.S. (2020). A deep learning based assistive system to classify COVID-19 face mask for human safety with YOLOv3. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, pp. 1-5. <https://doi.org/10.1109/ICCCNT49239.2020.9225384>
- [33] Dewi, C., Chen, R.C., Jiang, X., Yu, H. (2022). Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks. *PeerJ Computer Science*, 8: e943. <https://doi.org/10.7717/peerj-cs.943>
- [34] Dewi, C., Chen, R.C., Liu, Y.T., Liu, Y.S., Jiang, L.Q. (2020). Taiwan stop sign recognition with customize

- anchor. In Proceedings of the 12th International Conference on Computer Modeling and Simulation, pp. 51-55. <https://doi.org/10.1145/3408066.3408078>
- [35] Bose, S.R., Kumar, V.S. (2022). In-situ recognition of hand gesture via Enhanced Xception based single-stage deep convolutional neural network. *Expert Systems with Applications*, 193: 116427. <https://doi.org/10.1016/j.eswa.2021.116427>
- [36] Vrigkas, M., Kourfalidou, E.A., Plissiti, M.E., Nikou, C. (2022). Facemask: A new image dataset for the automated identification of people wearing masks in the wild. *Sensors*, 22(3): 896. <https://doi.org/10.3390/s22030896>
- [37] Ng, D.H.L., Sim, M.Y., Huang, H.H., Sim, J.X.Y., Low, J.G.H., Lim, J.K.S. (2021). Feasibility and utility of facemask sampling in the detection of SARS-CoV-2 during an ongoing pandemic. *European Journal of Clinical Microbiology & Infectious Diseases*, 40: 2489-2496. <https://doi.org/10.1007/s10096-021-04302-6>
- [38] Jiang, M., Fan, X., Yan, H. (2020). Retinamask: A face mask detector. <https://arxiv.org/abs/2005.03950v2>.
- [39] Dewi, C., Chen, R.C., Jiang, X., Yu, H. (2022). Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4. *Multimedia Tools and Applications*, 81(26): 37821-37845. <https://doi.org/10.1007/s11042-022-12962-5>
- [40] Wei, T., Chen, D., Zhou, W., Liao, J., Zhang, W., Yuan, L., Hua, G., Yu, N. (2022). E2Style: Improve the efficiency and effectiveness of StyleGAN inversion. *IEEE Transactions on Image Processing*, 31: 3267-3280. <https://doi.org/10.1109/TIP.2022.3167305>
- [41] Nowrin, A., Afroz, S., Rahman, M.S., Mahmud, I., Cho, Y.Z. (2021). Comprehensive review on facemask detection techniques in the context of COVID-19. *IEEE Access*, 9: 106839-106864. <https://doi.org/10.1109/ACCESS.2021.3100070>
- [42] ultralytics. (2020). Yolo V5. <https://github.com/ultralytics/yolov5>.
- [43] Wang, Z., Jin, L., Wang, S., Xu, H. (2022). Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system. *Postharvest Biology and Technology*, 185: 111808. <https://doi.org/10.1016/j.postharvbio.2021.111808>
- [44] Carrasco, D.P., Rashwan, H.A., García, M.Á., Puig, D. (2021). T-YOLO: Tiny vehicle detection based on YOLO and multi-scale convolutional neural networks. *IEEE Access*, 11: 22430-22440. <https://doi.org/10.1109/ACCESS.2021.3137638>
- [45] Kasper-Eulaers, M., Hahn, N., Berger, S., Sebulonsen, T., Myrland, Ø., Kummervold, P.E. (2021). Detecting heavy goods vehicles in rest areas in winter conditions using YOLOv5. *Algorithms*, 14(4): 114. <https://doi.org/10.3390/a14040114>
- [46] Li, Z., Tian, X., Liu, X., Liu, Y., Shi, X. (2022). A two-stage industrial defect detection framework based on improved-yolov5 and optimized-inception-resnetv2 models. *Applied Sciences*, 12(2): 834. <https://doi.org/10.3390/app12020834>
- [47] Dewi, C., Chen, R.C., Tai, S.K. (2020). Evaluation of robust spatial pyramid pooling based on convolutional neural network for traffic sign recognition system. *Electronics*, 9(6): 889. <https://doi.org/10.3390/electronics9060889>
- [48] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. (2017). Feature pyramid networks for object detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 936-944. <https://doi.org/10.1109/CVPR.2017.106>
- [49] Liu, S., Qi, L., Qin, H., Shi, J., Jia, J. (2018). Path aggregation network for instance segmentation. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 8759-8768. <https://doi.org/10.1109/CVPR.2018.00913>
- [50] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- [51] Larxel. (2020). Face Mask Detection. Kaggle. <https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>.
- [52] Mikolaj Witkowski. (2020). Medical Mask Dataset. Kaggle. <https://www.kaggle.com/vtech6/medical-masks-dataset>.
- [53] Córdova, M., Pinto, A., Hellevik, C.C., Alaliyat, S.A.A., Hameed, I.A., Pedrini, H., Torres, R.D.S. (2022). Litter detection with deep learning: A comparative study. *Sensors*, 22(2): 548. <https://doi.org/10.3390/s22020548>
- [54] Sun, X., Jia, X., Liang, Y., Yang, B., Wang, M., Chi, X. (2022). An improved Yolo-V5 network for defect detection of a boiler inner wall. Available at SSRN 4057058. <https://dx.doi.org/10.2139/ssrn.4057058>
- [55] Sun, X.M., Zhang, Y.J., Wang, H., Du, Y.X. (2022). Research on ship detection of optical remote sensing image based on Yolo V5. In *Journal of Physics: Conference Series*, 2215(1): 012027. <https://doi.org/10.1088/1742-6596/2215/1/012027>
- [56] Arcos-García, Á., Álvarez-García, J.A., Soria-Morillo, L.M. (2018). Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing*, 316: 332-344. <https://doi.org/10.1016/j.neucom.2018.08.009>
- [57] Yang, H., Chen, L., Chen, M., Ma, Z., Deng, F., Li, M., Li, X. (2019). Tender tea shoots recognition and positioning for picking robot using improved YOLO-V3 model. *IEEE Access*, 7: 180998-181011. <https://doi.org/10.1109/ACCESS.2019.2958614>
- [58] Yuan, Y., Xiong, Z., Wang, Q. (2016). An incremental framework for video-based traffic sign detection, tracking, and recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(7): 1918-1929. <https://doi.org/10.1109/TITS.2016.2614548>
- [59] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 157: 417-426. <https://doi.org/10.1016/j.compag.2019.01.012>