applied sciences

MDPI

Article

A Low-Cost Detail-Aware Neural Network Framework and Its Application in Mask Wearing Monitoring

Silei Cao *, Shun Long and Fangting Liao

College of Information Science and Technology, Jinan University, Guangzhou 510632, China; tlongshun@jnu.edu.cn (S.L.); liaoft@stu2021.jnu.edu.cn (F.L.)
* Correspondence: slaycao1998@stu2020.jnu.edu.cn

Abstract: The use of deep learning techniques in real-time monitoring can save a lot of manpower in various scenarios. For example, mask-wearing is an effective measure to prevent COVID-19 and other respiratory diseases, especially for vulnerable populations such as children, the elderly, and people with underlying health problems. Currently, many public places such as hospitals, nursing homes, social service facilities, and schools experiencing outbreaks require mandatory mask-wearing. However, most of the terminal devices currently available have very limited GPU capability to run large neural networks. This means that we have to keep the parameter size of a neural network modest while maintaining its performance. In this paper, we propose a framework that applies deep learning techniques to real-time monitoring and uses it for the real-time monitoring of mask-wearing status. The main contributions are as follows: First, a feature fusion technique called skip layer pooling fusion (SLPF) is proposed for image classification tasks. It fully utilizes both deep and shallow features of a convolutional neural network while minimizing the growth in model parameters caused by feature fusion. On average, this technique improves the accuracy of various neural network models by 4.78% and 5.21% on CIFAR100 and Tiny-ImageNet, respectively. Second, layer attention (LA), an attention mechanism tailor-made for feature fusion, is proposed. Since different layers of convolutional neural networks make different impacts on the final prediction results, LA learns a set of weights to better enhance the contribution of important convolutional layer features. On average, it improves the accuracy of various neural network models by 2.10% and 2.63% on CIFAR100 and Tiny-ImageNet, respectively. Third, a MobileNetv2-based lightweight mask-wearing status classification model is trained, which is suitable for deployment on mobile devices and achieves an accuracy of 95.49%. Additionally, a ResNet mask-wearing status classification model is trained, which has a larger model size but achieves high accuracy of 98.14%. By applying the proposed methods to the ResNet mask-wearing status classification model, the accuracy is improved by 1.58%. Fourth, a mask-wearing status detection model is enhanced based on YOLOv5 with a spatial-frequency fusion module resulting in a mAP improvement of 2.20%. Overall, this paper presents various techniques to improve the performance of neural networks and apply them to mask-wearing status monitoring, which can help stop pandemics.

Keywords: mask-wearing status monitoring; convolutional neural network; global average pooling; feature fusion; attention mechanism

check for updates

Citation: Cao, S.; Long, S.; Liao, F. A Low-Cost Detail-Aware Neural Network Framework and Its Application in Mask Wearing Monitoring. *Appl. Sci.* **2023**, *13*, 9747. https://doi.org/10.3390/app13179747

Academic Editor: Krzysztof Koszela

Received: 27 June 2023
Revised: 22 August 2023
Accepted: 25 August 2023
Published: 29 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Background and Content of the Study

The adoption of deep learning techniques in real-time monitoring can save a lot of manpower in various scenarios. For example, mask-wearing is a highly effective measure to prevent the pandemic, especially for vulnerable groups such as children, the elderly, and people with underlying health problems. Incorrect mask-wearing primarily refers to situations where the nose or mouth is exposed. It is vital to wear masks correctly to ensure maximum protection. Once before, the strict containment policies kept COVID-19-positive

cases at a minimum in public places, so incorrect mask-wearing might not have posed a significant impact. However, the change to a more lenient policy has increased the risk of exposure to infected individuals, making correct mask-wearing even more important. The requirements for mask-wearing vary across different public settings. In high-risk areas such as fever clinics in hospitals, strict adherence to mask-wearing protocols is essential due to a higher risk of virus transmission. Real-time monitoring of individuals who wear masks incorrectly or have them removed is pivotal. In other public places, occasional incorrect mask-wearing or temporary removal of masks may have a less noticeable impact. In such cases, prioritizing accuracy is critical to avoid unnecessary alerts that could disrupt the overall crowd experience. It is expensive for cloud-based devices to provide real-time monitoring; hence, off-cloud detection provided by terminal devices is more preferable. However, most of these low-cost devices currently available have very limited GPU capability, necessitating the use of neural networks with fewer parameters. The main challenge in mask-wearing monitoring lies in the accurate identification of individuals who have a small portion of their mouths or noses exposed while wearing masks. Therefore, it is necessary to introduce as few parameters as possible to make the model focus on detailed features.

There are two primary categories of mask-wearing detection methods. The first category utilizes well-established face detection techniques to locate the position of the faces. Subsequently, the face region was extracted and fed into a classifier network to determine if the mask is worn correctly. Alternatively, a direct one-stage object detection algorithm is used to simultaneously detect the face and assess the presence of a mask. However, since the face detection technique in the first category is well-developed, further performance improvement is hard to achieve. Therefore, this paper concentrates on enhancing the classifier network of the first category and exploring the second approach as shown in Figure 1.

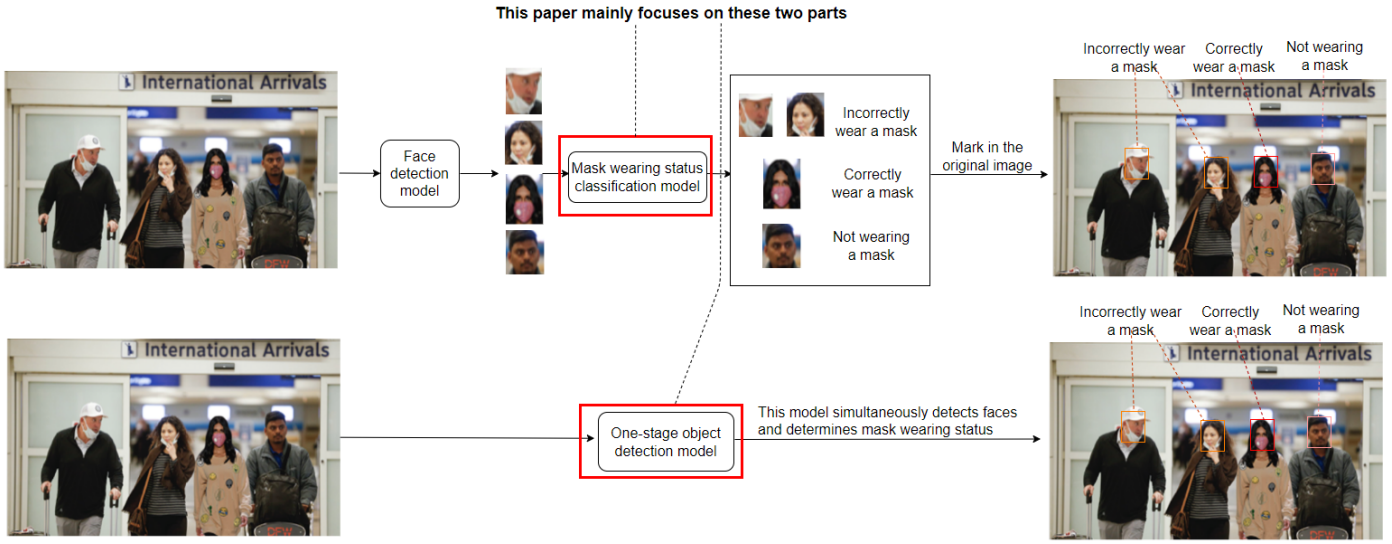


Figure 1. Two methods of mask wear detection.

1.2. Research Program

In this study, we developed a low-cost detail-aware neural network framework for real-time monitoring and applied it to mask-wearing status monitoring. It can effectively monitor mask compliance without human interference in situations where mask-wearing is compulsory. A significant amount of human resources can therefore be saved. The classification and detection of mask-wearing status are to be addressed in this research, corresponding to image classification and object detection tasks in computer vision respectively. Classification determines if the mask is correctly worn based on a given image with a single face. Detection means detecting the positions of all faces in the original image and telling whether they wear masks correctly. Given different emphases on real-time and accuracy for different scenarios, it is essential to select an appropriate model for each

task. For instance, in fever clinics where real-time detection is crucial (immediate alerts when someone is not wearing a mask), MobileNetv2 [1] is chosen for classification and YOLOv5 [2] is selected for detection. However, for scenarios such as elderly care where accuracy is prioritized (avoiding false recognitions that may disrupt the crowd experience), ResNet is preferable. Furthermore, novel modules such as the skip layer pooling fusion technique and layer attention mechanism are proposed and incorporated into ResNet to further improve accuracy. In addition, a YOLOv5 model with a spatial-frequency fusion module is utilized for detection in this specific scenario.

1.3. Major Contributions

Skip layer pooling fusion: It is observed that shallow layers possess more information of higher resolution about location and detail but offer less semantic understanding. On the contrary, deeper layers provide richer semantic information but may lack the perception of fine details. Therefore, the fusion of deep and shallow features is vital for improving the performance of mask-wearing classification models. Based on this idea, we introduce an innovative feature fusion technique named skip layer pooling fusion (SLPF) to fuse deep and shallow features in a neural network. In comparison to other feature fusion methods, this architecture effectively improves accuracy while reducing the complexity (number of parameters) associated with feature fusion.

Layer attention: We observe that the features extracted from each convolutional layer of the neural network contribute differently to the final prediction results. Some convolutional layers extract more important features, while others extract features that have a less noticeable impact on the prediction results. Based on this observation, we propose an attention mechanism called layer attention (LA) for feature fusion, aiming to prioritize the former whilst neglecting the latter.

Spatial-frequency feature fusion: The frequency domain features mainly include high-frequency and low-frequency components, which correspond to different features of the image. The features that reveal part of the nose and mouth are considered detail features of an image and produce abrupt changes in grey values. They correspond to high-frequency components in the frequency domain and are easier to analyze. To maximize the utilization of these features, a spatial-frequency feature fusion module was proposed.

1.4. The Structural Arrangement of This Paper

The Section 1 mainly introduces the research background, content, and main contributions of this article. The Section 2 introduces the relevant work of this article. The Section 3 first introduces the new method proposed in this article and then applies it to the classification and detection tasks of mask-wearing status. In the Section 4, we analyze the experimental results. The Section 5 summarizes the content of this article and points out its shortcomings and future research directions.

2. Related Work

2.1. Image Classification

The past decade has seen deep neural networks (DCNNs) make remarkable progress in speech, image, text, and other information-processing domains. They have emerged as fundamental components for image processing in numerous computer vision tasks. Various DCNN architectures have been proposed and successfully deployed in practice, for instance, ResNet [3] and others [4–11] tailored for image processing. However, a common practice in most DCNNs is to utilize deep features extracted solely from the final convolutional layer for prediction but overlook the significance of shallow features.

2.2. Feature Fusion

Prior studies [12–16] showed that the high-resolution shallow features in DCNN provide more information about location and detail but offer less semantic understanding. On the contrary, deeper features carry richer semantic information while being less perceptive

to finer details. Feature fusion techniques have been widely adopted in computer vision to enhance model performance, particularly through a combination of deep and shallow features to form pyramidal features. In object detection, networks such as YOLOv3 [17] or RetinaNet [18] employ feature pyramid networks (FPN) [19] for feature fusion. Two common fusion strategies have been adopted for semantic segmentation tasks, namely, high-resolution net (HRNet) [20] and FPN. They typically involve upsampling to align feature sizes, leading to additional parameters and increased computational complexity. However, challenges arise when applying feature fusion to image classification in a similar way to those in object detection and semantic segmentation. Specifically, the method first obtains deep feature maps with high-level semantic information but less detail by layer-by-layer convolution. This is followed by upsampling to match the sizes of deep and shallow feature maps. These feature maps are then fused with corresponding shallow feature maps that contain low-level semantic information and richer details. Finally, the fused features are flattened into a one-dimensional vector and fed into the fully connected layer for prediction. This approach significantly increases the number of parameters of the fully connected layer, which can account for 80% of the total network parameters. While the increased parameter cost is worthwhile in object detection and semantic segmentation tasks where precise location information is vital in final prediction, image classification primarily focuses on category information. In classification tasks, semantic features play a key role, with high-level semantic features being particularly important. On the other hand, low-level semantic features are relatively less effective and are used only as supplementary information. Therefore, DCNNs typically utilize features extracted from the last convolutional layer which contains high-level semantic features for prediction in classification. But only a small portion of shallow features containing low-level semantic and location information contributes to the final results. It is not cost-effective to introduce a large number of parameters to leverage the information from these shallow features that have far less impact on the final results. Hence, the key challenge in feature fusion for image classification networks lies in the effective utilization of shallow feature information while minimizing the number of parameters introduced by feature fusion.

2.3. Attention Mechanisms

Attention mechanisms such as channel attention [21–26], spatial attention [27–30], and hybrid attention [31–33] are commonly used in the literature. They primarily operate within individual layers of the neural network. However, they lack inter-layer connectivity across the attention module. Currently, feature fusion techniques have been adopted in various deep learning tasks, such as object detection and image segmentation. By combining deep and shallow features into pyramidal features, the overall performance of the model can be further enhanced. We believe that each layer of features has a different effect on prediction and therefore introduces a cross-layer attention mechanism.

2.4. Mask Wearing Detection

There are two primary approaches to mask-wearing detection. The first one utilizes face detection techniques to locate faces. The detected faces are then cropped and fed into a classifier to determine if the masks are worn correctly. Despite offering the advantage of reusing a pre-deployed face detection model and thus lowering the deployment costs, this method has a disadvantage in that established face detection techniques have all been optimized for uncovered faces and may not be able to detect faces covered with masks well. Vansh Gupta [34] employed a multi-task convolutional neural network (MTCNN) to segment face regions in images and used MobileNetv2 for mask-wearing classification. Alternatively, one-stage object detection algorithms are developed to detect faces and simultaneously detect and tell mask-wearing. Houkang Deng [35] enhanced the mask-wearing detection model by introducing inverse convolution and feature fusion to the original SSD model. Additionally, an attention mechanism was incorporated to filter and retain relevant information. These improvements significantly enhance the model's ability

to detect mask-wearing. Qing Ye [36] improved YOLOv4 for mask-wearing detection by integrating the CBAM attention module and substituting ordinary convolution with depthwise over-parameterized convolution. These methods can classify images into two categories: with or without masks, but cannot distinguish between correct and incorrect mask-wearing. Although Shuyi Guo [37] can tell between correct, incorrect, and no mask-wearing, they have not taken into account the impact of frequency domain features on the detection results, which is significant, as shown later in this paper.

3. Methodology

3.1. Overall Architecture

Figure 2 shows the training and usage process of the real-time classification model proposed. For better understanding, we show an example of mask-wearing classification, where the dataset is first uploaded to the cloud and enhanced (randomly cropped and rotated) to expand the dataset and then placed in a neural network model with SLPF and LA added and trained on the cloud. The resulting model is then deployed to the terminal device, where the input images are classified.

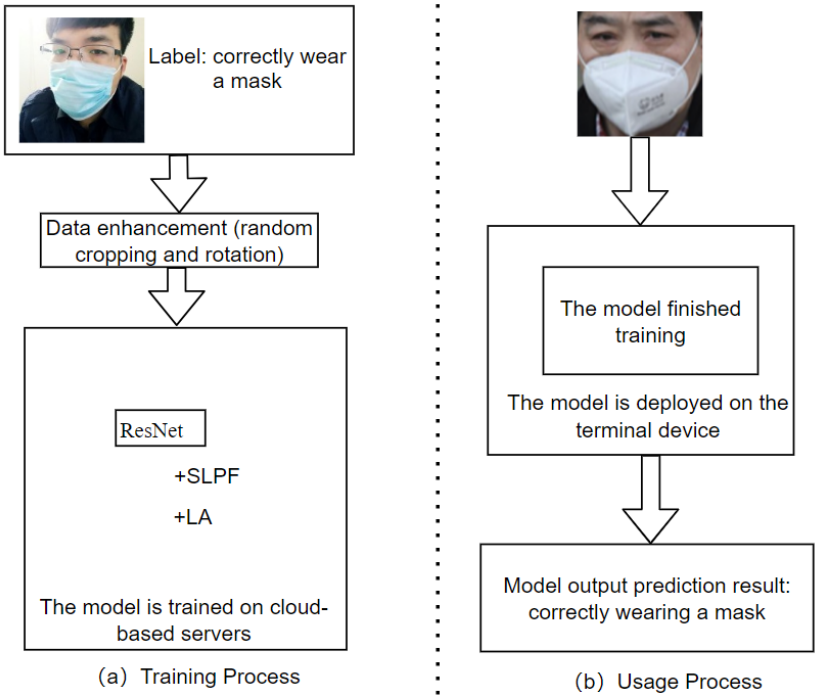


Figure 2. Training and usage process of the real-time classification model.

Figure 3 shows the training and usage process of the real-time detection model. For better understanding, we show the example of mask-wearing detection, where correct mask-wearing, incorrect mask wearing, and no mask-wearing correspond to label 0, label 1, and label 2 in the training set, respectively. The dataset is first uploaded to the cloud and enhanced to train YOLOv5 enhanced with spatial-frequency feature fusion proposed in this paper. After training, the resulting model is deployed to the terminal device, where the input images are analyzed.

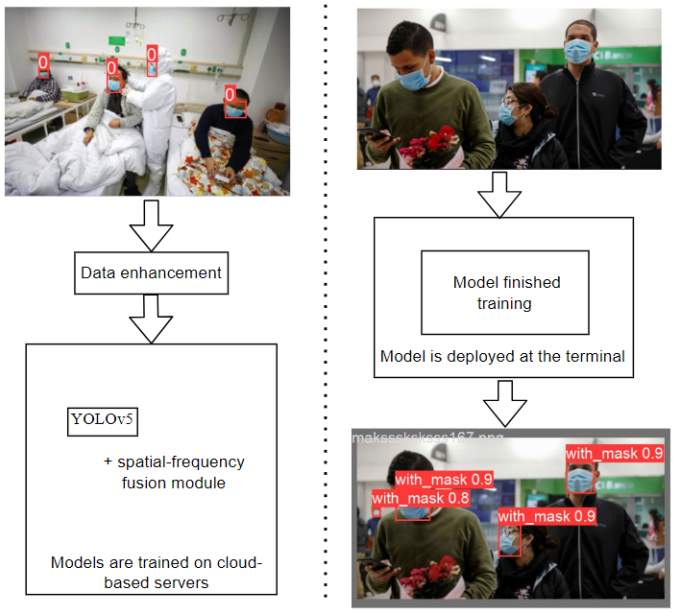


Figure 3. Training and usage process of the real-time detection model.

3.2. Skip Layer Pooling Fusion

We present a simple but effective feature fusion technique called skip layer pooling fusion in this subsection. Figure 4 illustrates the architecture of a conventional CNN for image classification, while Figure 5 depicts the SLPF architecture, where a branch is introduced at each intermediate layer of the neural network. The branches we added use global average pooling(GAP) [38] to align the features. After GAP, the size of the feature maps is the same (1×1 for all), and then the features are fused. It is worth noting that the critical parameter of skip layer pooling fusion technology is the convolutional kernel size of the convolutional layer added before global average pooling. In our experiment, we used 1×1, 3×3 and 5×5 convolutional kernels and then found that their effects were almost the same. While considering the parameter quantity, we chose a 1×1 convolutional kernel. The pseudocode of SLPF is shown in Algorithm 1.

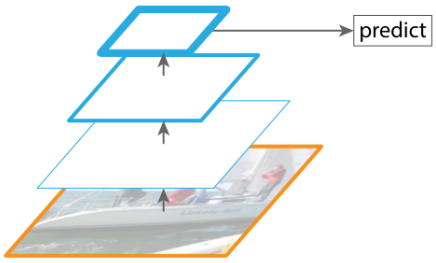


Figure 4. A regular CNN image classifier.

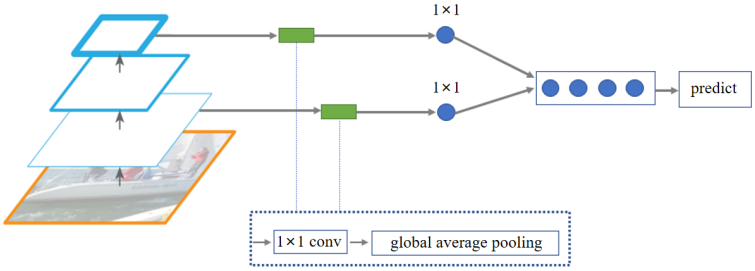


Figure 5. A CNN classifier with SLPF added.

Algorithm 1: Skip layer pooling fusion.

```
x = conv1(x) // Forward propagation, performing convolution operations,
    where x is the input image.
x1 = gap(conv1_finetune(x)) // Use global average pooling to compress
    features after fine-tuning them.
x = conv2(x) // Forward propagation, performing convolution
    operations.
x = conv3(x) // Forward propagation, performing convolution
    operations.
x3 = gap(conv3_finetune(x)) // Use global average pooling to compress
    features after fine-tuning them
x = torch.cat(x1,x3) // Feature fusion
x = conv_transfer(x) // Channel conversion
x = fc(x) // Feed the obtained features into the fully connected layer
```

To achieve this, a decision must be made as to which layers the branch should be added to. It depends on the last convolutional layer that does not contain a convolutional kernel size of 1×1 for feature extraction in CNN. If it is an odd layer, the branches are added to the odd layers. Otherwise, they are added to the even layers. This is because prior studies suggest that the features extracted by the last convolutional layer have the most significant impact on the classification results. That is why most CNNs use only the features from the last layer for classification. The motivation behind employing skip-layer feature fusion is that we think adjacent convolutional layers tend to extract similar features. By incorporating the features extracted from the skip layers, we can mitigate issues such as excessive model parameters and overfitting, resulting in an improvement of the model's overall performance.

The proposed SLPF structure offers several advantages. First, feature fusion is performed after the GAP of the features from each layer. This approach effectively compresses the features of the model before fusion. For instance, given an input image of size 256×256, if without utilizing GAP, as shown in Figure 6, the size of the feature maps obtained from layer B after multiple convolutions may be $32 \times 32 \times b$, where b is the number of convolutional kernels in layer B, while the size of the previous layer A could be $64 \times 64 \times a$, where a is the number of convolutional kernels in A. When the feature maps of B are upsampled and combined with the feature maps of A, the output becomes $64 \times 64 \times (a + b)$.

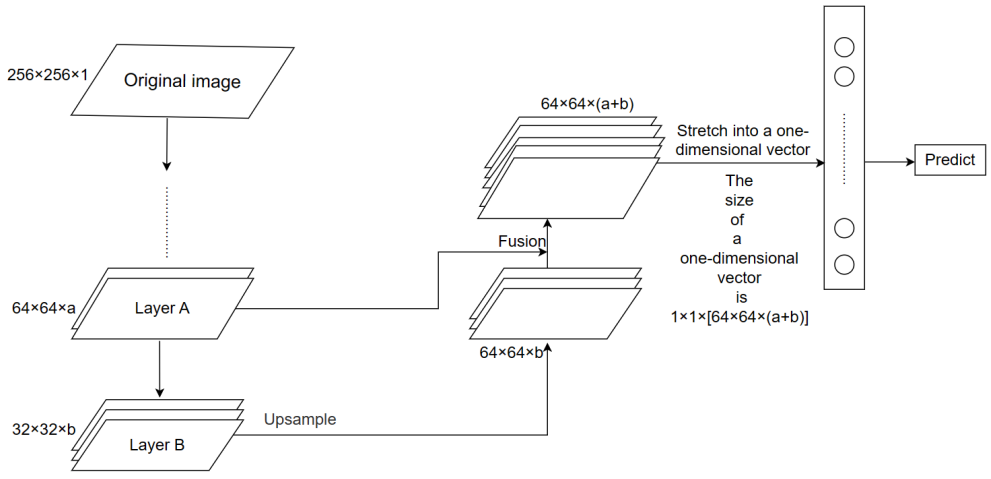


Figure 6. Common feature fusion methods using upsampling aligned features.

By utilizing GAP, the feature map sizes of A and B are reduced to $1 \times 1 \times a$ and $1 \times 1 \times b$, respectively, as shown in Figure 7. Consequently, the size of fused features will be $1 \times 1 \times (a + b)$, which is significantly smaller when compared to $64 \times 64 \times (a + b)$ (around 4000 times). This fusion method significantly reduces the number of parameters added to the model, i.e., the size of the model. Additionally, instead of directly fusing the features extracted from each convolutional layer, we fuse the features from the jump layer. This further reduces the number of parameters introduced by feature fusion and helps prevent overfitting.

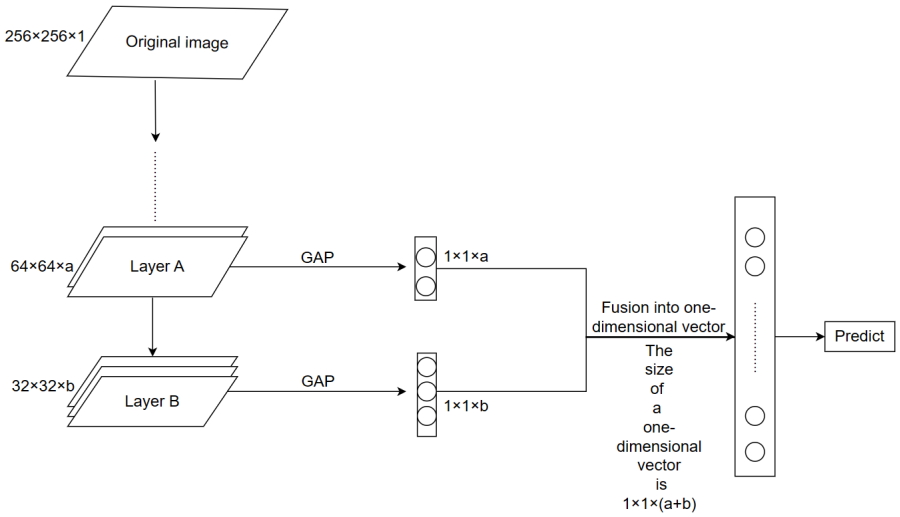


Figure 7. Our proposed feature fusion method.

Figure 8 shows how SLPF is integrated into an existing deep neural network model, with SqueezeNet as an example

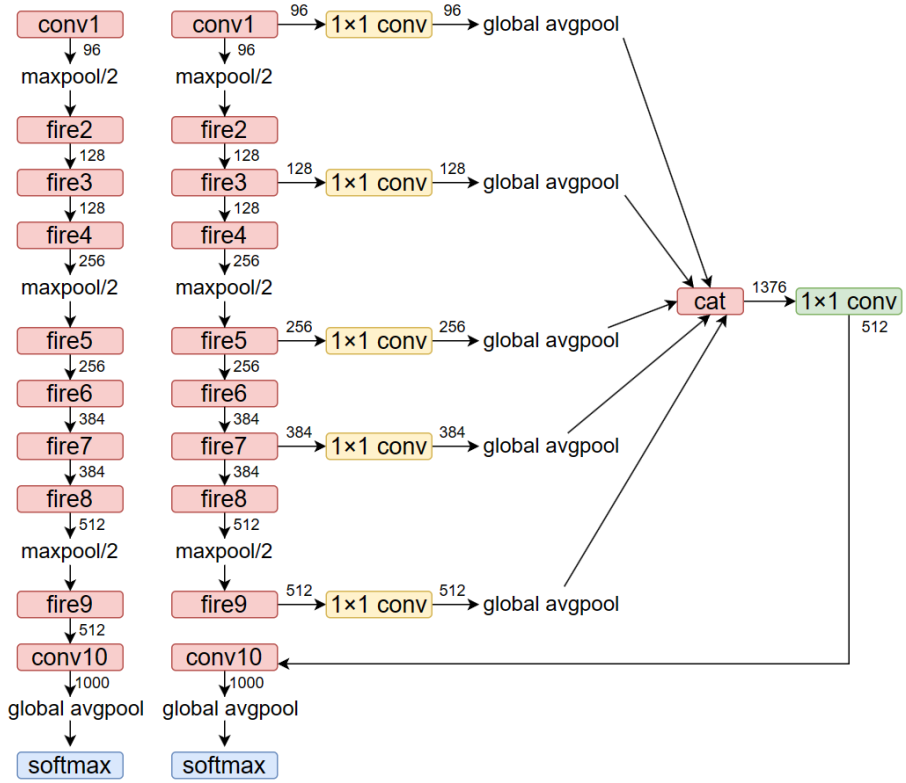


Figure 8. SqueezeNet (left) and SLPF-SqueezeNet (right).

It shows that SqueezeNet has ten convolutional layers and modules. Since conv10 is a convolutional layer with a convolutional kernel size of 1×1 for dimensionality reduction rather than feature extraction, fire9 is the last convolutional layer used for feature extraction. Therefore, SLPF introduces a branch in each odd layer to refine the features through a convolutional layer. Subsequently, GAP is employed to align the deep and shallow features. Once aligned, SLPF performs feature fusion and uses a 1×1 convolutional layer to map the number of channels to the original network's channel number. This step plays a crucial role in preserving the parameters of the original network structure.

3.3. Layer Attention

The proposed layer attention mechanism is explained in depth in this subsection with its architecture shown in Figure 9. To compress the global spatial information in a deep convolutional neural network, we introduce a GAP operation after each convolutional layer and convolutional module. Let us assume the input feature map has dimensions $H_i \times W_i \times C_i$, where H_i, W_i, and C_i represent the height, width, and number of channels of the feature map at the i-th layer, respectively. First, the feature map is compressed from size $H_i \times W_i \times C_i$ to $1 \times 1 \times C_i$ through GAP. Then, we fuse the $1 \times 1 \times C_i$ feature maps obtained from the compressed part of each layer along the channels. The fused vector has a size of $1 \times 1 \times (C_1 + C_2 + C_3 + \ldots + C_n)$, where n represents the number of convolutional layers or modules to be fused. Next, we pass the fused vector through the fully connected layer to predict the contribution of each convolutional layer or module. The output consists of n numbers, representing the contribution of each layer's feature to the final prediction result. The weights obtained from this prediction are then multiplied by feature maps of each layer to obtain the weighted feature maps of them, as shown in Figure 10. The weighted feature maps are subsequently used for feature fusion. Finally, they are fed into a classification or regression layer to make the prediction. The pseudocode of LA is shown in Algorithm 2.

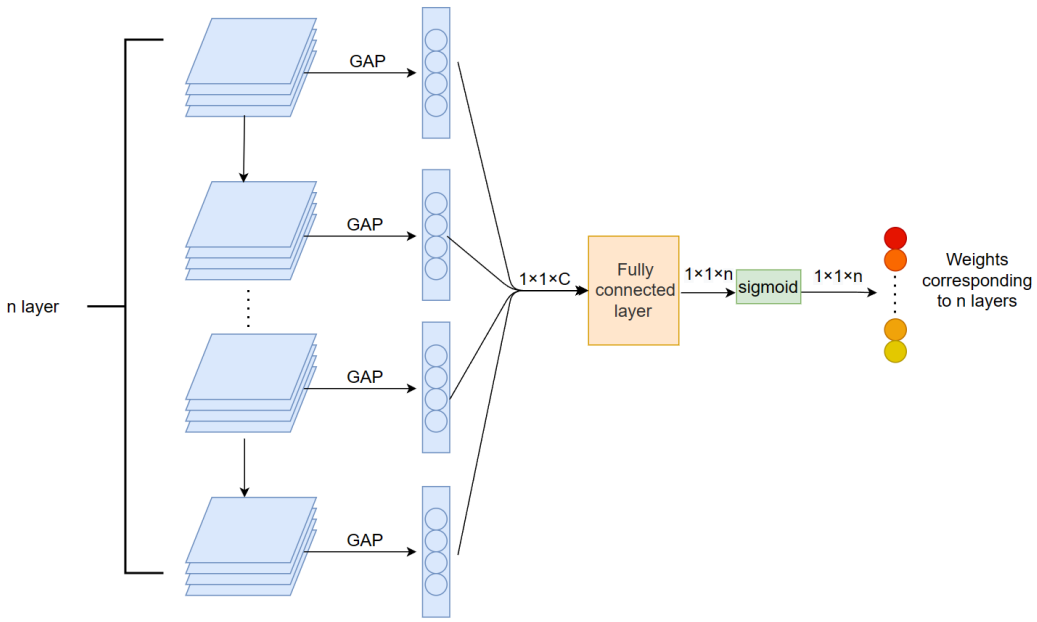


Figure 9. Schematic diagram of the layer attention mechanism to predict the weight of each layer.

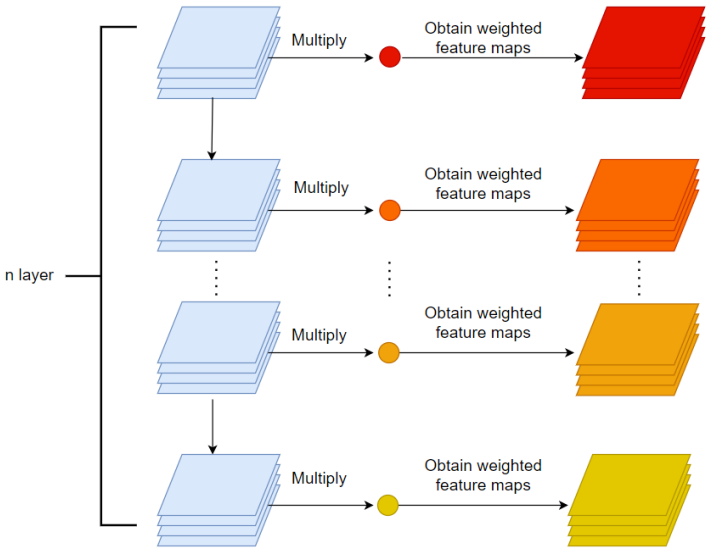


Figure 10. Weighted feature map obtained by multiplying the predicted weights of each layer.

Algorithm 2: Layer attention.

```
x1 = conv1(x) // Forward propagation, performing convolution operations, where x is the input image.
gap1 = gap(x1) // Utilizing global average pooling to compress features.
x2 = conv2(x1) // Forward propagation, performing convolution operations.
gap2 = gap(x2) // Utilizing global average pooling to compress features.
x3 = conv3(x2) // Forward propagation, performing convolution operations.
gap3 = gap(x3) // Utilizing global average pooling to compress features.
x = torch.cat(gap1,gap2,gap3) // Feature fusion
weight = sigmoid(fc_predict_weight(x)) // Place the obtained features into the fully connected layer to predict weights.
weighted_features = (x1,x2,x3) × weight // The original feature map is multiplied by the corresponding weights to obtain a weighted feature map.
```

3.4. Mask Wearing Classification

ResNet and MobileNetv2 are chosen for our mask-wearing status classification. Despite having a larger model size and slower classification speed, ResNet offers higher accuracy, which is appealing when storage space is sufficient and real-time requirements are not compulsory. On the contrary, MobileNetv2 has a smaller model size and faster detection speed, but its accuracy is lower when compared to ResNet. It is preferable when storage space is limited and real-time is a must. To further enhance the performance of the models, data augmentation techniques such as random cropping and rotation have been applied to the dataset. For MobileNetv2, we deliberately avoided adopting additional techniques to maintain its lightweight and keep the model size modest. In addition, to better strengthen the performance of ResNet, SLPF was introduced to effectively utilize features from its different layers. This leads to a notable performance improvement. Finally, LA was incorporated into the model to prioritize relevant information in the search for

enhanced performance. The overall schematic diagram of the neural network incorporating both SLPF and LA is presented in Figure 11.

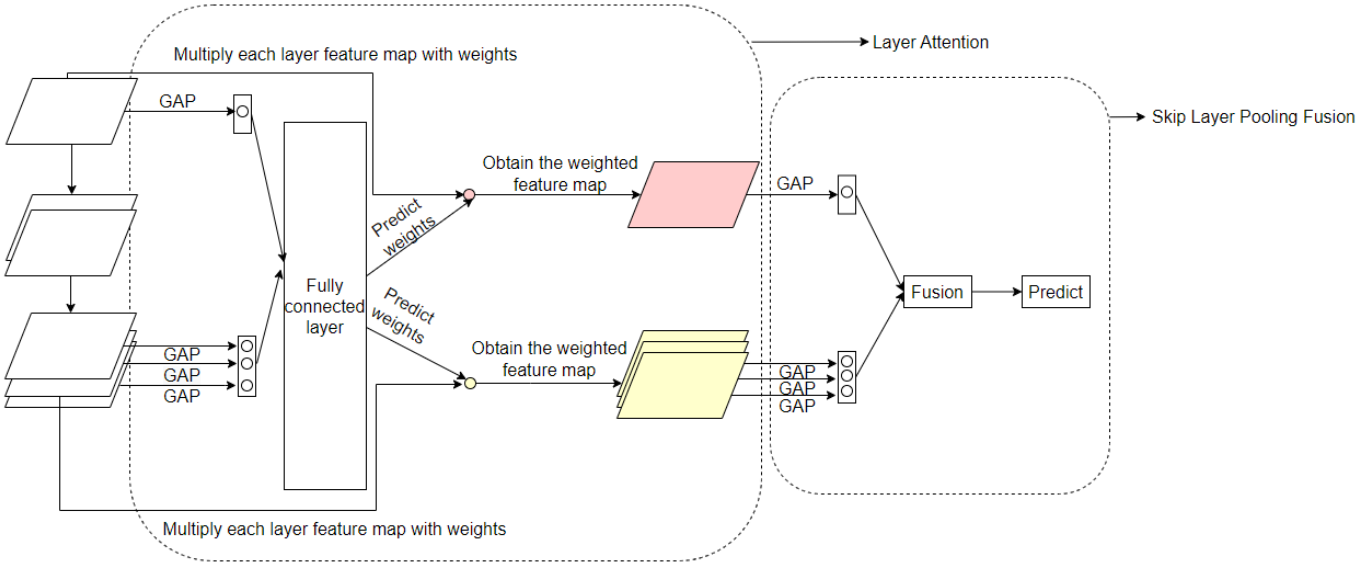


Figure 11. Schematic diagram of the network architecture with both SLPF and LA added.

3.5. Mask Wearing Detection

A fast one-stage YOLOv5s object detection model was introduced as the foundation for mask-wearing status detection. Additionally, a spatial-frequency feature fusion module was added to enhance its performance, and its pseudocode is shown in Algorithm 3. We believe that the features that reveal part of the nose and mouth are detail features of the image that is associated with abrupt changes in grey values, i.e., high-frequency components in the frequency domain. To maximize the utilization of image features, a spatial-frequency feature fusion module was proposed. The main concept of the spatial-frequency feature fusion module involves applying the Fourier transformation to the extracted features to convert them into frequency domain features before feeding them to the classifier. The result of Fourier transformation contains two parts: the real and the imaginary. Since the latter predominantly consists of zeros, this paper employs only the real part of the fused frequency domain features with the spatial features for prediction to avoid introducing redundant parameters. The spatial and frequency domain features are then combined and sent to the classifier for prediction. The architecture of the whole model is shown in Figure 12.

Algorithm 3: Spatial-frequency feature fusion.

```
x = conv1(x) // Forward propagation, performing convolution operations,
    where x is the input image.
x = conv2(x) // Forward propagation, performing convolution
    operations.
x = conv3(x) // Forward propagation, performing convolution
    operations.
fft1 = fft(x) // Perform Fourier transform on features.
x = torch. cat(x,fft1.real) // Fusing the real part of frequency domain
    features with the original features.
x = fc(x) // Feed the obtained features into the fully connected layer.
```

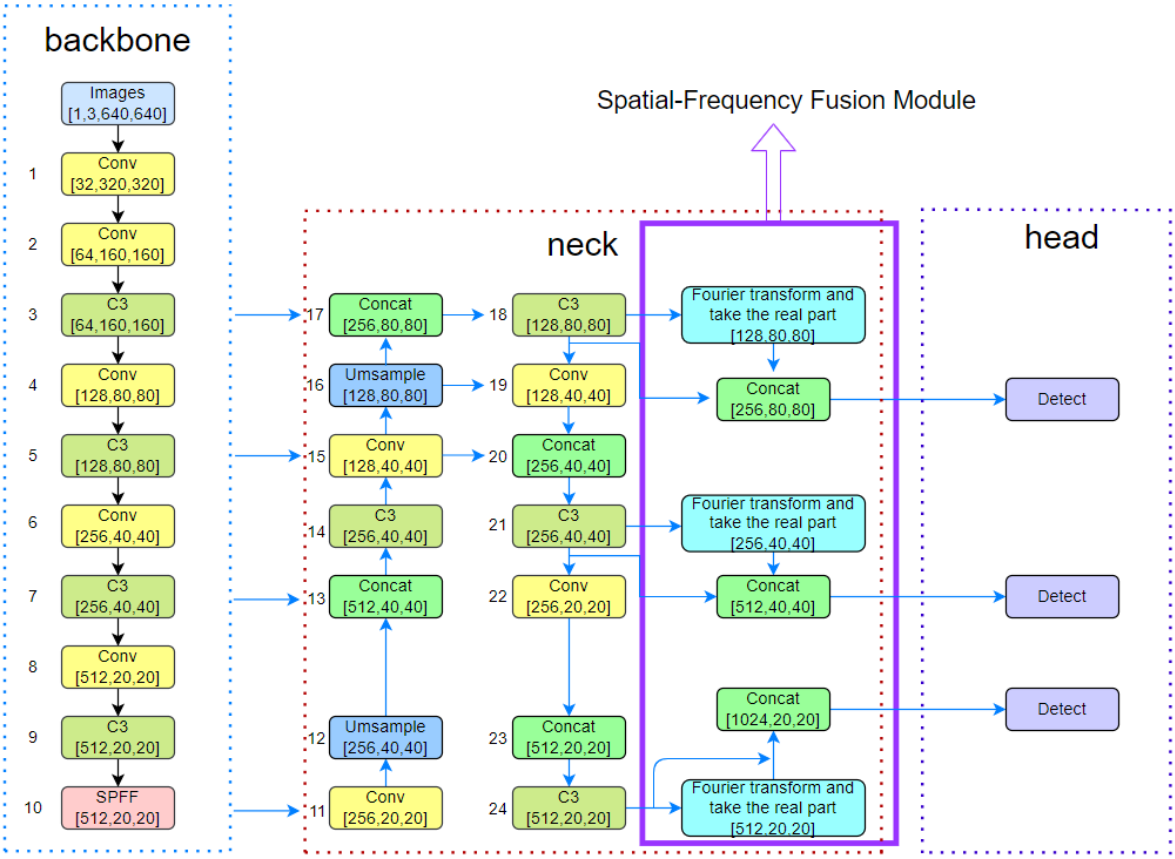


Figure 12. The YOLOv5 model with the spatial-frequency fusion module.

4. Experimental Results and Analysis

4.1. Dataset and Experiment Setup

We adopted various datasets in our experiment to evaluate the effectiveness of our proposed model. In addition, we conducted five cross-validation experiments on each dataset with a ratio of 5:1 between the training and validation sets, and the experimental results were averaged.

To evaluate the effectiveness of SLPF and LA, we conducted experiments on Cifar100 [39] and Tiny-ImageNet [40] datasets. The experiments were carried out on a hardware platform with an NVIDIA GeForce RTX 2080Ti. The initial learning rate was set to 0.1, with a batch size of 128. We applied a weight decay of 5×10^{-4} and a Nesterov momentum of 0.9 during the training process, which lasted for 200 epochs. The learning rate was reduced by a factor of 5 at the 60th, 120th, and 160th epochs.

To evaluate the ability of our mask-wearing status classification model, we collected 3399 photos of correctly worn masks, 3298 photos of incorrectly worn masks, and 3913 photos of no masks. The parameter settings were the same as for the ablation experiments evaluating SLPF and LA. In addition, we deployed the model on Android phones and conducted real-time monitoring testing, with the device collecting 1583 photos for testing. Android operating environment configuration: Redmi K30pro with 8 GB of running memory and Qualcomm Snapdragon 865 processor.

To evaluate the ability of our mask-wearing status detection model, it was first trained with a set of 853 face photos capturing individuals wearing masks from various angles. The experiments were carried out on a hardware platform with an NVIDIA GeForce RTX 2080Ti for conducting the trials. Stochastic gradient descent was employed as the optimizer with an initial learning rate of 0.01, a weight decay of 0.01, and the Nesterov momentum set to 0.95. The models underwent training for a duration of 300 epochs.

4.2. Ablation Experiments

The results of the ablation experiments are summarized in Table 1, which demonstrates the performance improvements achieved by introducing SLPF and LA in all the deep learning models used in our experiments. In general, SLPF led to accuracy improvements ranging from 2.10% to 9.65% for all DCNNs. The average accuracy improvements it brought are 4.78% and 5.21% on CIFAR100 and Tiny-ImageNet, respectively. Furthermore, LA contributed to accuracy improvements ranging from 1.44% to 3.57% for all DCNNs. On average, it led to accuracy improvements of 2.10% and 2.63% on CIFAR100 and Tiny-ImageNet, respectively. In addition, it is observed that both SLPF and LA brought more performance improvement to SqueezeNet, MobileNetv2, and ShuffleNetv2 models than on other models. This is mainly because these three models are lightweight models with fewer parameters, which inherently limits their learning capability. By incorporating SLPF and LA, these models can effectively utilize the features from each layer, thereby strengthening their learning capability.

Table 1. Experimental results of multiple models on CIFAR100 and Tiny-ImageNet.

	Cifar100 Accuracy (%)	**Tiny-ImageNet Accuracy (%)**	**The Improvement of Accuracy Compared to the Previous Model on Cifar100 and Tiny-ImageNet**
SqueezeNet	70.28	58.15	
SLPF-SqueezeNet	75.84 (+5.56)	63.76 (+5.61)	7.91% and 9.65%
LA-SLPF-SqueezeNet	77.39 (+1.55)	65.68 (+1.92)	2.04% and 3.01%
MobileNetv2	68.55	56.09	
SLPF-MobileNetv2	72.38 (+3.83)	60.16 (+4.07)	5.59% and 7.26%
LA-SLPF-MobileNetv2	74.25 (+1.87)	62.31 (+2.15)	2.58% and 3.57%
ShuffleNetv2	70.25	60.35	
SLPF-ShuffleNetv2	74.26 (+4.01)	62.40 (+2.05)	5.71% and 3.40%
LA-SLPF-ShuffleNetv2	76.40 (+2.14)	64.23 (+1.83)	2.88% and 2.93%
ResNet18	76.18	64.29	
SLPF-ResNet18	78.15 (+1.97)	66.12 (+1.83)	2.59% and 2.85%
LA-SLPF-ResNet18	79.37 (+1.22)	67.41 (+1.29)	1.56% and 1.95%
GoogleNet	76.77	66.34	
SLPF-GoogleNet	78.38 (+1.61)	68.27 (+1.93)	2.10% and 2.91%
LA-SLPF-GoogleNet	79.51 (+1.13)	69.44 (+1.17)	1.44% and 1.71%

To present how SLPF and LA make effects during the process, we plotted the performance of SqueezeNet against the increasing epoch during training, and the results are shown in Figures 13 and 14. Figure 13 displays that at roughly 60, 120, and 160 epochs, the learning rate drops and the accuracy increases abruptly. If the learning rate remains constant, the loss may oscillate around its minimum. Once the learning rate is reduced, the amplitude of oscillation keeps decreasing until reaches its minimum, i.e., the highest accuracy. However, decreasing the learning rate more frequently will keep the loss at its local minimum rather than reaching the global optimum. Therefore, we choose to decrease the learning rate only at 60, 120, and 160 epochs. It can be seen from Figures 13 and 14 that in the early stages of training, the model has not yet extracted very good features from the deep layers of the convolutional neural network. At this time, the shallow features have a significant impact on the final prediction. Therefore, there is a huge gap in accuracy and loss value between the model with added SLPF and the original model. As the number of training times increases, better features are extracted from deep layers; thus, the gap between the two gradually narrows. However, the shallow layer still contains features that have an impact on the prediction results and also do not exist in the deep layer, so there is still a significant gap. In addition, we can also see that the model with LA added did not

learn the weight parameters very well in the early training stage, enhancing the features that have little impact on the prediction results and reducing the feature layers that have a significant impact on the prediction results. Hence, the accuracy of the model with LA added is almost always lower than that of the model with only SLPF added. As the epoch increases, the model learns the appropriate parameters. Therefore, the accuracy is higher and the loss value is lower than the model with only SLPF added.

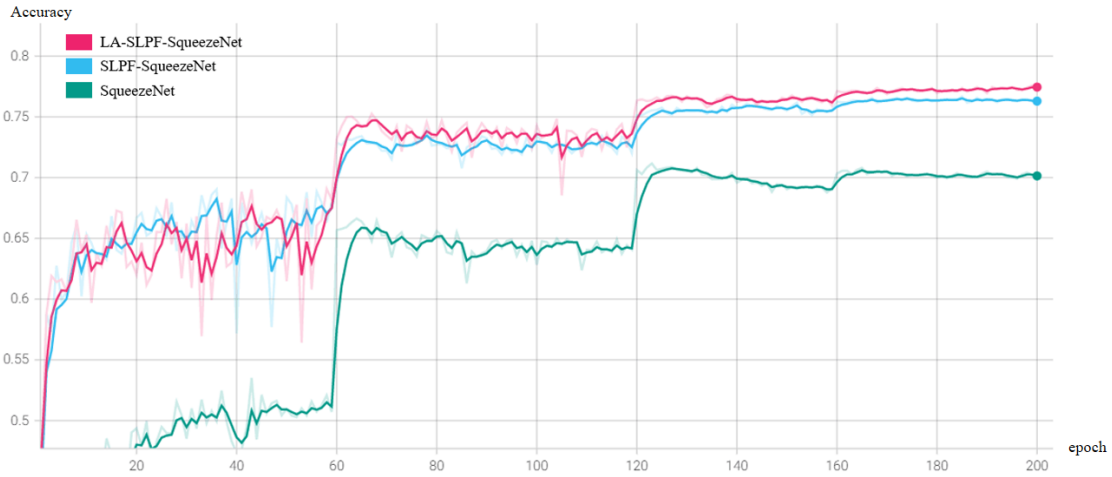


Figure 13. SqueezeNet, SLPF-SqueezeNet, and LA-SLPF-SqueezeNet accuracy change curves in the test set.

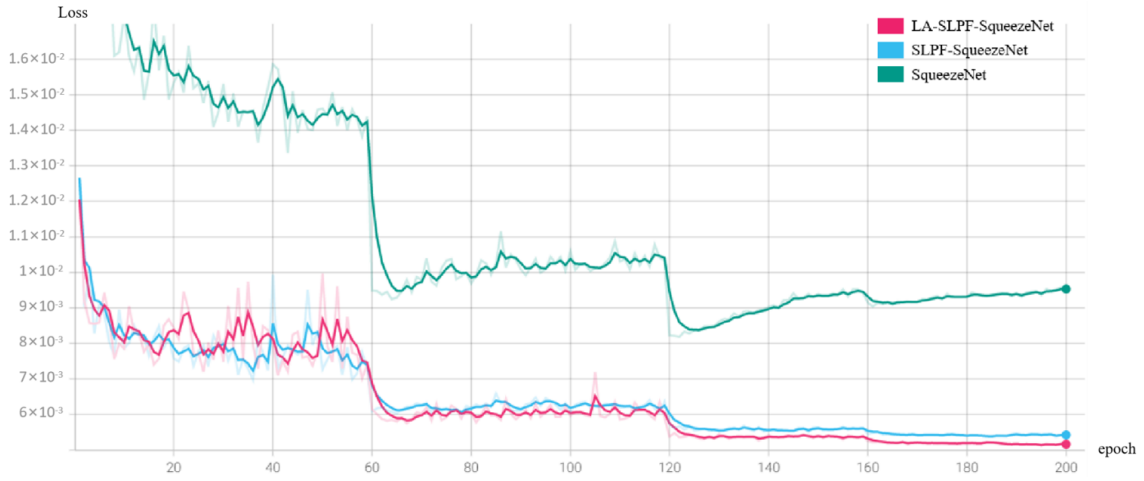


Figure 14. SqueezeNet, SLPF−SqueezeNet, and LA−SLPF−SqueezeNet loss change curves in the test set.

In addition, we also compared LA with the most commonly used attention mechanisms of SE and CBAM. The experimental results are shown in Table 2. It can be seen that the attention mechanisms of SE and CBAM can better improve the accuracy of the model than the LA proposed in this paper, but they introduce much larger parameter quantities than LA. This is because the SE attention mechanism learns a weight for each feature map, while LA shares a weight for feature maps in the same layer. Assuming a convolutional neural network has 1024 feature maps in a specific layer, SE needs to predict 1024 weights, while LA only needs to predict one weight, and all feature maps in that layer share that weight. The CBAM mixed attention mechanism not only includes channel attention mechanisms similar to SE but also introduces spatial attention mechanisms, so it introduces more parameters than SE.

Table 2. Experimental results obtained by adding different attention mechanisms.

Model	Accuracy (%)	Number of Parameters Introduced
SLPF-ResNet18	78.15	
+SE	79.68 (+1.96%)	242 KB
+CBAM	79.74 (+2.03%)	252 KB
+LA	79.37 (+1.56%)	8 KB

4.3. Mask Wearing Classification

To better understand the model training process, we plotted the variations in model accuracy against the training iteration, and the results are shown in Figure 15. It shows that our proposed model consistently outperformed the original model during the training phase. To be more specific, it reveals that MobileNetv2 attained an accuracy of 95.49%, while ResNet achieved an accuracy of 96.61%. Upon incorporating SLPF into ResNet, the accuracy was improved to 97.50%. Moreover, by adding both SLPF and LA to the ResNet model, the accuracy was further increased to 98.14%. These results suggest that the proposed techniques led to a 1.58% enhancement in accuracy for ResNet. It can be seen from Figure 15 that the accuracy of the model with added SLPF and LA is almost always higher than that of the model with only added SLPF, and the accuracy of the model with only added SLPF is always higher than the original model. Unlike Figure 13, the accuracy of the model with SLPF and LA added at the beginning is already higher than that of the model with only SLPF added. This is mainly because the model is easy to learn the rules for identifying the wearing status of masks. Therefore, in the early stages of training, LA effectively enhances the feature layers that have a significant impact on the prediction results.

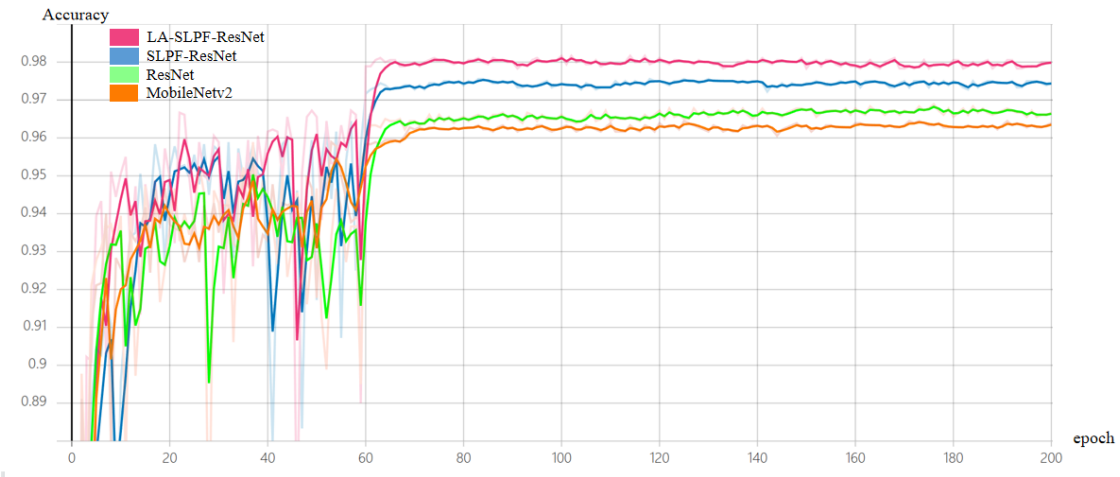


Figure 15. Multiple models' accuracy change curves on the test set.

4.4. Mask Wearing Detection

Table 3 presents the average detection accuracy(AP@0.5) of the original YOLOv5 model for all three categories: incorrect mask-wearing, correct mask-wearing, and no mask. The corresponding AP@0.5 scores for these categories are 66.72%, 92.04%, and 73.28%, respectively. The mean average precision (mAP@0.5) is 77.35%. Once the spatial-frequency fusion module was incorporated into YOLOv5, AP@0.5 was improved to 71.36%, 92.22%, and 73.56% for all three categories, respectively, and mAP@0.5 was increased to 79.05%. A comparison of the original YOLOv5 and its variant suggests improvements in AP@0.5 were 6.95%, 0.20%, and 0.38%, respectively, and mAP@0.5 increased by 2.20%. These results highlight the significant contribution the spatial-frequency fusion module has made to the overall performance, particularly in spotting incorrect mask-wearing. However,

the introduction of frequency domain features resulted in a decrease in FPS. Therefore, when emphasizing real-time performance, we can use the original YOLOv5s, and when emphasizing accuracy, we can use the model proposed in this article.

Table 3. Experimental results of the original model and our model.

	AP@0.5 (%)			mAP@0.5 (%)	FPS
	Mask_Weared_Incorrect	With_Mask	No_Mask		
original model	66.72	92.04	73.28	77.35	105
our model	71.36 (+6.95%)	92.22 (+0.20%)	73.56 (+0.38%)	79.05 (+2.20%)	98

Figure 16 represents the PR curve for the YOLOv5 model, while Figure 17 represents the PR curve for the YOLOv5 model enhanced with spatial-frequency fusion. In these figures, the yellow, light blue, green, and dark blue lines correspond to the PR curves for no, correct, incorrect mask-wearing, and overall, respectively. The values at the top-right corner are the AP@0.5 scores for each category.

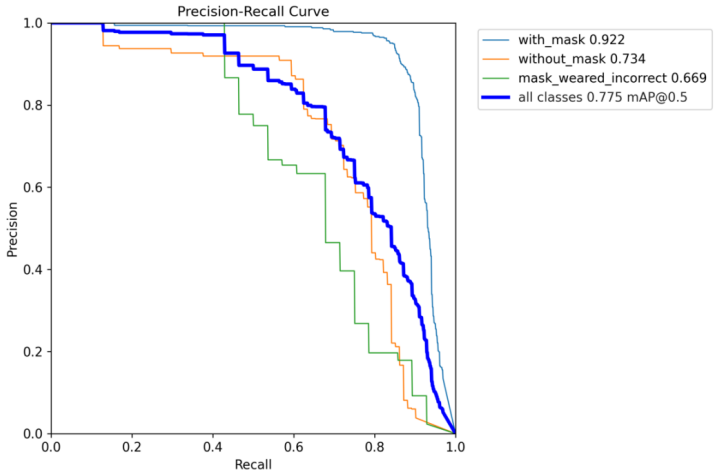


Figure 16. PR curve graph of the YOLOv5 model.

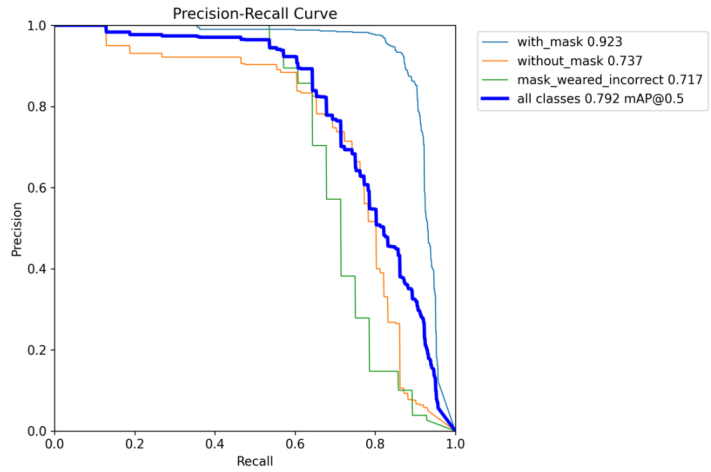


Figure 17. PR curve graph of the YOLOv5 model with the spatial-frequency fusion module added.

Figures 18 and 19 demonstrate the changes in various metrics, including loss values, precision, recall, mAP@0.5, and mAP@0.95 during training for both the YOLOv5 model and its enhancement with the spatial frequency fusion. In these figures, "train" and "val" represent the training and the validation set, respectively. The red box in the graphs

indicates a sudden drop in precision and a corresponding rise in recall at around the 80th epoch of training. This behavior stems from the initial high number of bounding boxes set by YOLOv5 for predictions. As the number of iterations increases, the number of bounding boxes decreases and object localization becomes more accurate, resulting in a sudden drop in accuracy while recall rises.

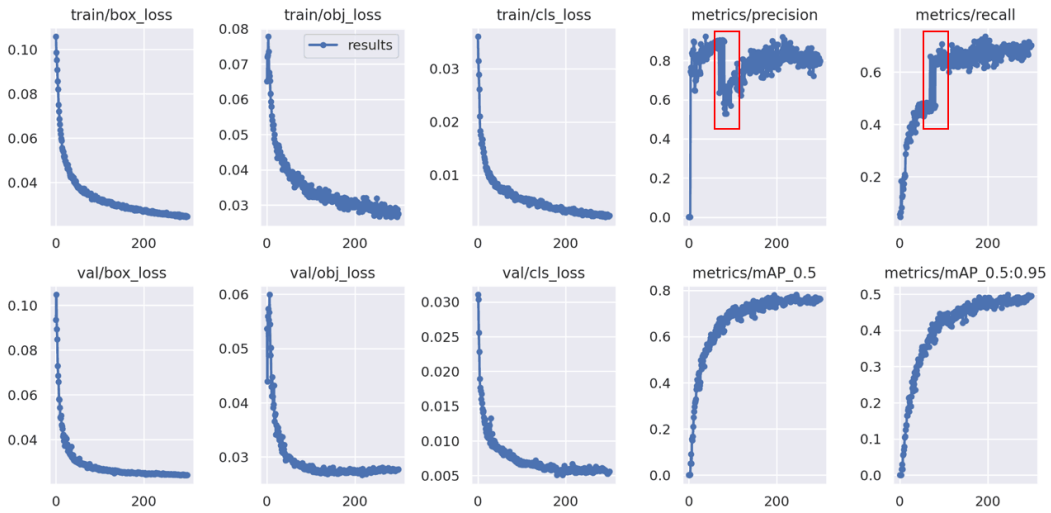


Figure 18. Training curve of the original YOLOv5 model.

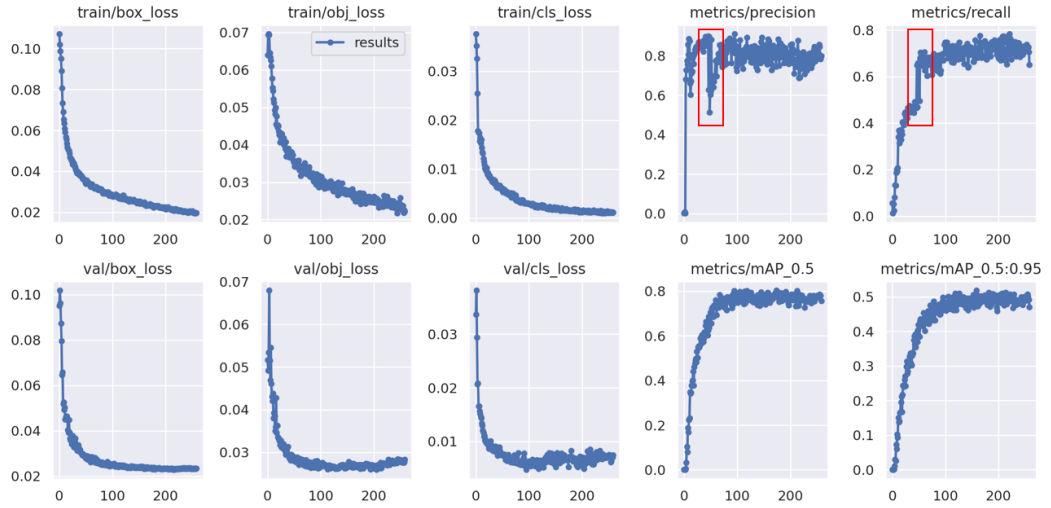


Figure 19. Training curve graph of the YOLOv5 model with the spatial-frequency fusion module added.

We deployed the obtained model on mobile devices and tested it. The device has collected 1583 photos for testing, and the experimental results are shown in Table 4. It can be seen that the model mAP@0.5 is 80.31% and FPS is 22.

Table 4. The detection results of the model deployed on the Android device.

	AP@0.5 (%)		**mAP@0.5 (%)**	**FPS**
Mask_Weared_Incorrect	**With_Mask**	**No_Mask**		
73.58	91.94	75.42	80.31	22

Figures 20–22 show one original image, its corresponding detection results of YOLOv5, and that of its enhancement with spatial-frequency fusion, respectively. Figure 21 shows that people who are not wearing a mask correctly are misidentified as wearing a mask correctly (indicated by the green box in Figure 21). However, Figure 22 demonstrates that the proposed model rectifies the misclassification by correctly identifying them as not correctly wearing a mask. These results demonstrate the performance enhancement ability of the proposed method.



Figure 20. Original image.

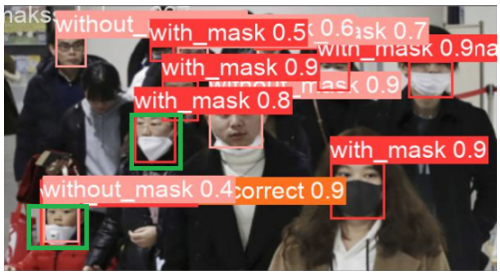


Figure 21. Detection results of YOLOv5.

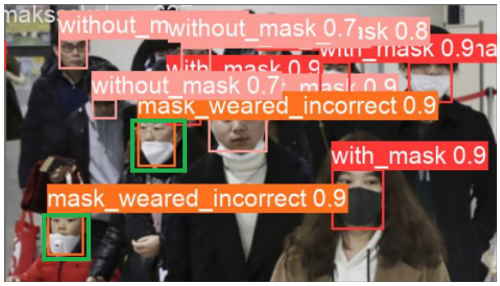


Figure 22. Detection results of the YOLOv5 enhanced with the spatial-frequency fusion module.

5. Conclusions

This study puts forward a framework for real-time monitoring using deep learning technology and applies it to the real-time monitoring of mask-wearing status. To achieve this goal, we introduce two innovative methods. Firstly, SLPF is designed to address the drawback of most convolutional neural networks currently used for classification, which only use the feature maps of the last layer for classification. It is worth noting that compared to the upsampling-based method, our method prominently reduces the number of parameters involved and the computational cost of feature fusion. Overall, for various DCNNs, SLPF resulted in a boost in accuracy between 2.10% and 9.65%. On average, SLPF yielded an evidently-enhanced accuracy of each DCNN by 4.78% and 5.21% on CIFAR100 and Tiny-ImageNet, respectively. Secondly, we developed LA to tackle the different contributions of each layer when combining features from each layer. LA learns a set of weights, allowing the model to prioritize feature layers that make a significant contribution to the final prediction while diluting the impact of less informative layers. Extensive experiments have proven the effectiveness of LA, and for various DCNNs, the accuracy of LA has been promoted by 1.44% and 3.57%. On average, it has increased by

2.10% and 2.63% on CIFAR100 and Tiny ImageNet, respectively. In addition, MobileNetv2 and ResNet are selected for mask-wearing classification in diverse scenarios. MobileNetv2 achieved an accuracy of 95.49%, while ResNet achieved 98.14%. By applying the proposed enhancement, the ResNet mask-wearing status classification model culminated in a substantial accuracy increase by 1.58%. YOLOv5 was used for mask-wearing detection. Intending to maximize the utilization of image features, a spatial frequency fusion module was proposed, effectively boosting the performance of the model. The experimental results reveal that the YOLOv5 model strengthened with spatial-frequency fusion achieves an AP@0.5 improvement of 6.95% when detecting incorrect mask wearing, and mAP@0.5 is increased by 2.20%.

Nevertheless, it is essential to acknowledge the limitations inherent in the model demonstrated in this study. For example, for the model to perceive detailed features, we have added many branches, but some of them may be redundant, thereby increasing the complexity of the model. So, the next step is to prune and quantify the model to further compress its size and deploy it in practical applications.

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

Funding: This work has been partly supported by the Joint Research Fund in Astronomy under a cooperative agreement between the National Natural Science Foundation of China and the Chinese Academy of Sciences (No. U2031104), Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011340).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Research data are not shared due to confidentiality reasons.

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

References

1. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
2. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part IV 14; Springer: Cham, Switzerland, 2016; pp. 630–645.
5. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
6. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
7. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
8. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
9. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
10. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *PMLR, Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019*; MLResearch: Tempe, AZ, USA, 2019; pp. 6105–6114.
11. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.

12. Sun, L.; Chen, J.; Xie, K.; Gu, T. Deep and shallow features fusion based on deep convolutional neural network for speech emotion recognition. *Int. J. Speech Technol.* **2018**, *21*, 931–940. [CrossRef]
13. Zeng, C.; Zhu, D.; Wang, Z.; Yang, Y. Deep and shallow feature fusion and recognition of recording devices based on attention mechanism. In *Advances in Intelligent Networking and Collaborative Systems: Proceedings of the 2th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2020) 12*; Springer: Cham, Switzerland, 2021; pp. 372–381.
14. Han, L.; Zhao, Y.; Lv, H.; Zhang, Y.; Liu, H.; Bi, G. Remote sensing image denoising based on deep and shallow feature fusion and attention mechanism. *Remote Sens.* **2022**, *14*, 1243. [CrossRef]
15. Yue, X.; Chen, X.; Zhang, W.; Ma, H.; Wang, L.; Zhang, J.; Wang, M.; Jiang, B. Super-resolution network for remote sensing images via preclassification and deep–shallow features fusion. *Remote Sens.* **2022**, *14*, 925. [CrossRef]
16. Wang, Z.; Yang, Y.; Zeng, C.; Kong, S.; Feng, S.; Zhao, N. Shallow and deep feature fusion for digital audio tampering detection. *EURASIP J. Adv. Signal Process.* **2022**, *2022*, 69. [CrossRef]
17. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
18. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
19. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
20. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5693–5703.
21. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
22. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7151–7160.
23. Lee, H.; Kim, H.E.; Nam, H. Srm: A style-based recalibration module for convolutional neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1854–1862.
24. Yang, Z.; Zhu, L.; Wu, Y.; Yang, Y. Gated channel transformation for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11794–11803.
25. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
26. Qin, Z.; Zhang, P.; Wu, F.; Li, X. Fcanet: Frequency channel attention networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 783–792.
27. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks. *arXiv* **2015**, arXiv:1506.02025.
28. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
29. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-excite: Exploiting feature context in convolutional neural networks. *arXiv* **2018**, arXiv:1810.12348.
30. Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C.C.; Lin, D.; Jia, J. Psanet: Point-wise spatial attention network for scene parsing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 267–283.
31. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
32. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
33. Roy, A.G.; Navab, N.; Wachinger, C. Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018, Proceedings of the 21st International Conference, Granada, Spain, 16–20 September 2018*; Proceedings, Part I; Springer: Cham, Switzerland, 2018; pp. 421–429.
34. Gupta, V.; Rajput, R. Face mask detection using MTCNN and MobileNetV2. *Int. Res. J. Eng. Technol. (IRJET)* **2021**, *8*.
35. Deng, H.; Zhang, J.; Chen, L.; Cai, M. Improved mask wearing detection algorithm for SSD. *J. Phys. Conf. Ser.* **2021**, *1757*, 012140. [CrossRef]
36. Ye, Q.; Zhao, Y. Mask wearing detection algorithm based on improved YOLOv4. *J. Phys. Conf. Ser.* **2022**, *2258*, 012013. [CrossRef]
37. Guo, S.; Li, L.; Guo, T.; Cao, Y.; Li, Y. Research on Mask-Wearing Detection Algorithm Based on Improved YOLOv5. *Sensors* **2022**, *22*, 4933. [CrossRef]
38. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
39. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
40. Le, Y.; Yang, X. Tiny imagenet visual recognition challenge. *CS 231N* **2015**, *7*, 3.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.