Mastering the Mask: Harnessing Ensemble Learning for Defect Detection

Cheng-Yuan Ho
Department of Information Management
National Taiwan University
Taipei City, Taiwan
tommyho@ntu.edu.tw

Jia-En Wang
EXPETECH CO., LTD.
Taichung City, Taiwan
rtfgvb74125@gmail.com

Abstract—**The outbreak of the COVID-19 pandemic in 2019 led to a significant increase in the number of people wearing masks. As the demand for masks soared, numerous manufacturers entered the mask production industry. However, traditional quality control and defect detection processes in mask manufacturing heavily relied on manual inspection, resulting in substantial manpower requirements and time-consuming procedures. Although the implementation of AI-powered visual inspection aided in categorizing defects, distinguishing certain intricate flaws remained challenging, with an average model accuracy of 89.5% and an error rate of 10.5%. To address this limitation, this study adopts an ensemble learning approach, utilizing multiple models for prediction and employing majority voting to determine the final outcome. The ensemble learning technique successfully reduces the model's error rate by 16.2%, indicating its effectiveness in mitigating errors and enhancing accuracy. Nevertheless, solely relying on ensemble learning does not sufficiently lower the error rate. Thus, this study further investigates model ensemble by adjusting model weights to assess whether accuracy can be further improved. Experimental results demonstrate that with weight adjustment, the error rate can be minimized to as low as 7%, representing a two-thirds decrease from the current average error rate. This substantial improvement signifies the potential of weight adjustment in enhancing mask defect detection accuracy.**

Keywords—ensemble learning, defect detection, deep learning, mask, smart manufacturing

I. Introduction

At the end of 2019, the COVID-19 virus broke out and spread globally, infecting hundreds of millions of people and causing millions of deaths. In order to avoid infection, people have taken various protective measures, such as getting vaccinated, practicing frequent handwashing, and wearing masks. Among these measures, wearing masks is one of the most common practices, as masks effectively isolate viruses in the air and reduce the risk of infection [1]. As a result, the demand for masks has surged, and factories have been working tirelessly to produce more masks for the public.

During the mask production process, defects may inevitably occur, rendering some masks unusable, reducing their protective functions, or resulting in incomplete appearances. Currently, most mask factories rely on manual inspection for quality control, which not only takes a considerable amount of time but also incurs significant labor costs. Therefore, there is a desire to integrate artificial intelligence, specifically deep learning methods, to expedite the mask inspection process. The aim is to leverage ensemble learning techniques to enhance defect detection and minimize inspection time.

Deep learning (DL) is a subset of machine learning (ML), which in turn is a subset of Artificial Intelligence (AI). Therefore, deep learning can be considered as one of the methods within the field of AI. Both machine learning and deep learning utilize large datasets to enable machines to mimic human brain functions and learn reasoning and classification for problem-solving. The main difference between the two lies in their learning approach [1].

In machine learning, feature extraction is a prerequisite step. Taking cats and dogs as an example, assuming that the distinguishing features are fur color, ear shape, and face structure, these features are first extracted and then fed into the training model. On the other hand, deep learning does not require manual feature extraction. It directly inputs data into the model for training, and the model autonomously extracts features and makes judgments. Deep learning allows computer networks consisting of multiple layers of neural networks to have multiple abstract data representations, significantly enhancing breakthroughs in image recognition, speech recognition, and other fields.

In this research, we employ the Convolutional Neural Network (CNN) approach for defect detection and classification of masks in image recognition [2]. For model selection, we opt for the EfficientNets series as our primary training model. EfficientNets were introduced by Google in 2019, proposing a novel neural network scaling method that balances the network's depth, width, and image resolution. Here, depth refers to the number of layers in the neural network, width pertains to the number of channels in the network, and image resolution denotes the size of the image input into the neural network. Through the balance of these three dimensions, the network expansion becomes more effective [3]. Additionally, Google employed the Neural Architecture Search (NAS) algorithm to obtain a new baseline network, which was then scaled to produce a series of models known as EfficientNets [3].

We chose EfficientNets as our primary model due to their outstanding performance on ImageNet compared to other models. For instance, EfficientNet-B4 improved the top-1 accuracy from 76.3% to 83.0% compared to the widely used ResNet-50, with reduced computational complexity. Moreover, EfficientNet-B7 achieved a top-1 accuracy of 84.3% on ImageNet, with the model being 8.4 times smaller and 6.4 times faster [3].

However, during the model training process, it was observed that a single EfficientNet model did not perform as expected in mask defect recognition. To enhance the model's accuracy, we decided to employ ensemble learning [4], which involves multiple models classifying an image and then voting to determine the final classification result using a majority vote approach. We aim to investigate whether this ensemble learning method can effectively improve the accuracy of the model in image classification.

Research supported by the National Science and Technology Council (NSTC), Taiwan, under Grant Nos. MOST 109-2221-E-002-213-MY3 and NSTC 112-2221-E-002-003.

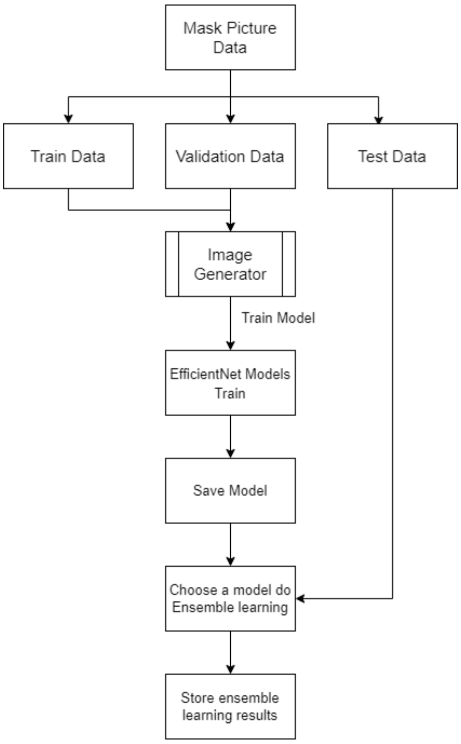
979-8-3503-1584-4/23/$31.00 ©2023 IEEE

II. Research Methods

In this section, we will outline the steps involved in our research. Firstly, masks are categorized manually into seven main classes, and the detailed descriptions of each class will be provided in the subsequent chapter. Next, the categorized masks are photographed and stored using the Insta 360 One R action camera. Subsequently, all the images are divided into three datasets - Train, Validation, and Test - following a 6:2:2 ratio. Once the data splitting is complete, data augmentation and preprocessing are applied to the Train and Validation datasets using an image generator.

After data preprocessing, we use EfficientNet models from B0 to B6, and each model undergoes transfer learning using the Train and Validation datasets. The trained models are then saved for later use. Finally, employing the Ensemble learning method, an odd number of models are selected, and the Test dataset is used for validation. We observe whether the Ensemble learning approach can enhance the accuracy of the models in classifying masks. Figure 1 illustrates the whole research workflow.

Fig. 1. Research workflow.


Mask Picture Data
Train Data
Validation Data
Test Data
Image Generator
Train Model
EfficientNet Models Train
Save Model
Choose a model do Ensemble learning
Store ensemble learning results

III. Dataset Introduction

The mask data used in this study can be categorized into seven main classes, as follows:

1. Normal: Masks without any defects, fully functional for regular use. (Figure 2a)

2. Ear Strap Defect: Masks with incomplete or detached ear straps during production. (Figure 2b)

3. Missing Ear Straps: Masks with ear straps incompletely attached or missing on both sides due to insufficient bonding or failure to replenish during production. (Figure 2c)

4. Nose Bridge Defect: Masks with incorrectly placed nose bridge strips during ultrasonic sealing or with nose bridge strips cut too short, resulting in incomplete sealing by the ultrasonic machine and potential damage to the machine. (Figure 2d)

5. Missing Nose Bridge Strip: Masks with incomplete sealing during production, causing the nose bridge strip to come off, or forgetting to replenish nose bridge strips during manufacturing. (Figure 2e)

6. Uneven Ear Strap Welding: Masks with securely attached ear straps but with different heights on both sides due to rotation during the attachment process, causing discomfort to the wearer. (Figure 2f)

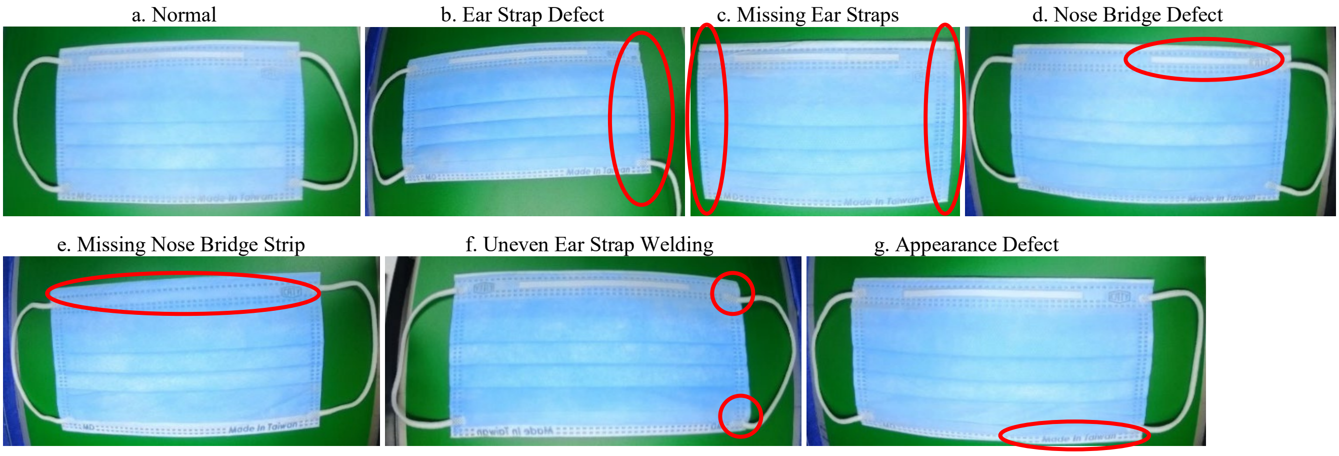
7. Appearance Defect: Masks with cosmetic imperfections, mainly seen on the upper and lower edges of the mask, resulting in uneven fabric appearance. However, this defect does not affect the mask's protective performance. (Figure 2g)

After manually categorizing the masks into the above seven classes, detailed documentation of the quantity of each category is provided in Table I. Once all masks have been photographed and documented, the images are split into Train, Validation, and Test datasets following a 6:2:2 ratio. The Train and Validation datasets are primarily used for model training, while the Test dataset is used for validating the ensemble learning models.

TABLE I. Mask Classes and Their Respective Quantities

Class	Class ID	No. (pics)
Normal	Normal	2058
Ear Strap Defect	Error_Ear	1562
Missing Ear Straps	No_Ear	1935
Nose Bridge Defect	Error_Iron	2017
Missing Nose Bridge Strip	No_Iron	440
Uneven Ear Strap Welding	Error_Solder	1934
Appearance Defect	NG	2036

Fig. 2. Seven main classes in mask dataset.


a. Normal
b. Ear Strap Defect
c. Missing Ear Straps
d. Nose Bridge Defect
e. Missing Nose Bridge Strip
f. Uneven Ear Strap Welding
g. Appearance Defect

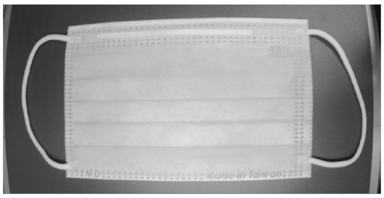
IV. Data Processing

The data preprocessing in this study is divided into two parts. Firstly, there are image format processing steps that are applied to all three datasets, i.e., Train, Validation, and Test datasets. Secondly, specific data augmentation techniques are performed exclusively on the Train Data. The following will provide separate explanations for these two approaches.

A. Image Format Processing

Firstly, before model training, all images in the three datasets need to undergo color conversion, resolution resized, and value normalization. To reduce the impact of different colors of masks on the model's performance, all mask images, which were collected in various colors and quantities, are converted from colored RGB images to grayscale images. This transformation reduces the importance of color during model learning as shown in Figure 3.

Fig. 3. RGB to Grayscale Conversion.



Regarding resolution, the original images captured during mask photography are of size 1920 x 1080. Training the model with such large images would place a significant burden on the GPU due to increased resource requirements. To ensure smooth model training while preserving crucial feature information, as shown in Table II, three different image sizes were tested for training and validation, with the size 240 x 140 performing the best. Consequently, all images were uniformly resized to 240 x 140.

TABLE II. Resolution Selection

Image Resolution	Classification Accuracy
120 x 70	0.83534
240 x 140	0.91037
360 x 210	0.88995

Lastly, image value normalization is performed to scale the pixel values in the image matrix between 0 and 1. This step is implemented to highlight the differences between images and eliminate similarities that may exist between them, as perceived by the machine's vision.

B. Data Augmentation

Large amounts of data are essential for effective deep learning model training. During data collection, some mask classes, such as "Missing Nose Bridge Strip" and "Uneven Ear Strap Welding," are challenging to acquire, resulting in a smaller quantity of data compared to other classes. To ensure accurate recognition of these categories and reduce overfitting during training [5, 6, 7, 8], data augmentation is applied to the Train Data. This involves performing vertical and horizontal adjustments on the data within the dataset. Figure 4 illustrates an example of vertical and horizontal adjustments.

Fig. 4. An Example of Data Augmentation.

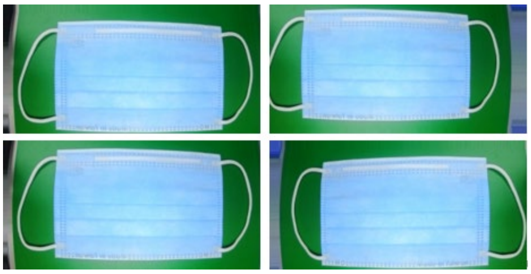
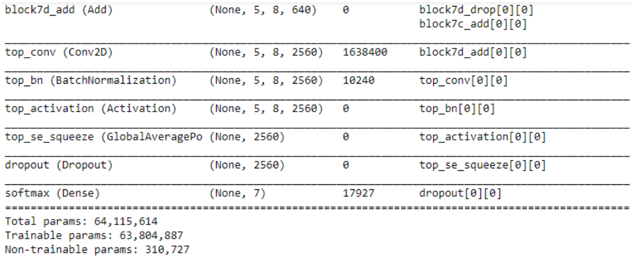


Fig. 5. Adjustment of EfficientNet's Dense Layer Neurons.



```
block7d_add (Add)              (None, 5, 8, 640)    0          block7d_drop[0][0]
                                                               block7c_add[0][0]
__________________________________________________________________________________
top_conv (Conv2D)              (None, 5, 8, 2560)   1638400    block7d_add[0][0]
__________________________________________________________________________________
top_bn (BatchNormalization)    (None, 5, 8, 2560)   10240      top_conv[0][0]
__________________________________________________________________________________
top_activation (Activation)    (None, 5, 8, 2560)   0          top_bn[0][0]
__________________________________________________________________________________
top_se_squeeze (GlobalAveragePo (None, 2560)        0          top_activation[0][0]
__________________________________________________________________________________
dropout (Dropout)              (None, 2560)         0          top_se_squeeze[0][0]
__________________________________________________________________________________
softmax (Dense)                (None, 7)            17927      dropout[0][0]
==================================================================================
Total params: 64,115,614
Trainable params: 63,804,887
Non-trainable params: 310,727
```

V. Training and Testing of EfficientNet Models

In the model training phase, this study introduces a total of eight EfficientNet models, ranging from B0 to B7, and trains them on the mask data. Before commencing training, slight adjustments are made to each model [9, 10, 11]. The original Dense layer outputs 1,000 classification neurons, which are modified to only have 7 neurons for classifying the mask categories, as there are only 7 mask classes as displayed in Figure 5.

After configuring the neural layers of the models, callback functions are set for the training process. Three callback functions, namely ModelCheckpoint, ReduceLROnPlateau, and EarlyStopping, are employed to serve different purposes. ModelCheckpoint is responsible for saving the best-performing model, comparing the performance after each iteration, and overwriting the previous model if the current performance is better. ReduceLROnPlateau defines the learning rate, reducing it during training when the model reaches a point where further significant improvements are unlikely. EarlyStopping is used to stop model training early when the model's performance plateaus, indicating no further improvement is expected.

Once all parameters are set, the models are trained. After completion, the performance of each model in each iteration is recorded in Table III. The loss of each model converges well, and the accuracy consistently improves, indicating increasing precision. In the ideal scenario, the validation data's loss and accuracy should closely approach the results of the Train data, demonstrating positive progress in model training. However, this chart does not fully represent the overall accuracy of the models. Therefore, further testing using the Test data is required for a comprehensive evaluation.

After completing the testing phase, Table IV contains the accuracy of each model and the corresponding confusion matrix for model classification. The results show that the accuracy of each model falls approximately between 0.87 and 0.90, with only the EfficientB0 model achieving an accuracy of 0.91. The average accuracy across all models reaches 89.5%, with an error rate of 10.5%. However, it is noteworthy that the classification errors in the "Error_Solder" category are particularly severe. Figure 6 shows the confusion matrix for each model's classification results.

To address the classification errors and enhance the overall performance, the study aims to leverage ensemble learning methods. By combining the predictions from multiple models, it is hoped that the ensemble approach will yield improved results for defect detection in masks.

TABLE III. MODELS' ITERATIONS

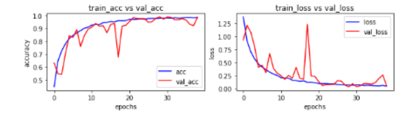
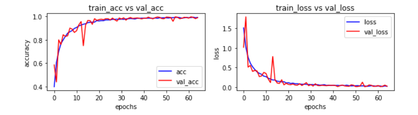
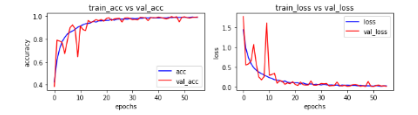
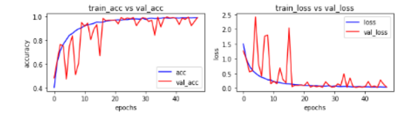
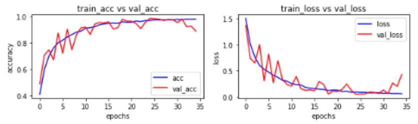
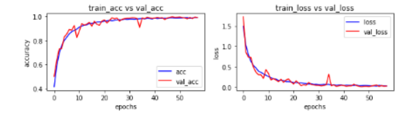
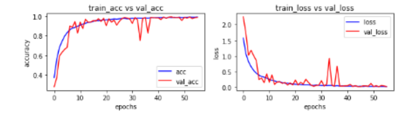
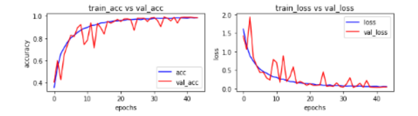
Model Name	Corresponding Iterations
EfficientNetB0	
EfficientNetB1	
EfficientNetB2	
EfficientNetB3	
EfficientNetB4	
EfficientNetB5	
EfficientNetB6	
EfficientNetB7	

TABLE IV. EFFICIENTNETS' ACCURACIES

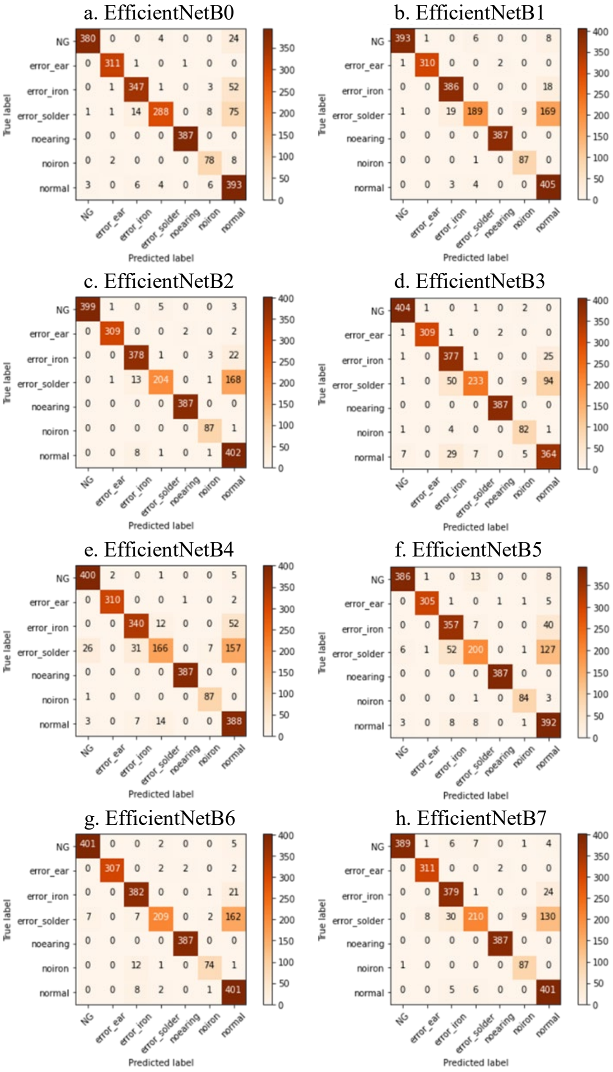
Model Name	Accuracy	Model Name	Accuracy
EfficientNetB0	0.910379	EfficientNetB4	0.866194
EfficientNetB1	0.899124	EfficientNetB5	0.879949
EfficientNetB2	0.902876	EfficientNetB6	0.900792
EfficientNetB3	0.898707	EfficientNetB7	0.902042

VI. ENSEMBLE MODEL PREDICTIONS

After completing all EfficientNets' training and testing, the selection of models for ensemble learning is based on their respective test accuracies. In this study, a total of 6 ensemble learning models are constructed, each comprising different combinations of EfficientNet models. The two crucial steps in model composition are as follows: first, selecting models with better performance for ensemble; second, ensuring that the number of models in each combination is odd. The reason for choosing models with better performance is to leverage their strengths to improve overall accuracy, and using an odd number of models avoids potential tie situations during voting, which could lead to random selection of a result. Thus, all ensemble models are composed of an odd number of constituent models as shown in Table V.

From Table V, it can be observed that the ensemble learning models achieve a slight improvement in test accuracy. The average accuracy of individual models was 89.5%, with an average error rate of 10.5%. However, through ensemble learning, the average accuracy is increased to 91.7%, with a 16.2% decrease in error rate. Though the improvement is modest, it is worth noting that the correct classifications in the "Error_Solder" category have significantly increased, and even the accuracy of the "Normal" category has slightly improved as shown in Figure 7.

Fig. 6. Confusion Matrix for Each Model's Classification Results.



a. EfficientNetB0

b. EfficientNetB1

c. EfficientNetB2

d. EfficientNetB3

e. EfficientNetB4

f. EfficientNetB5

g. EfficientNetB6

h. EfficientNetB7

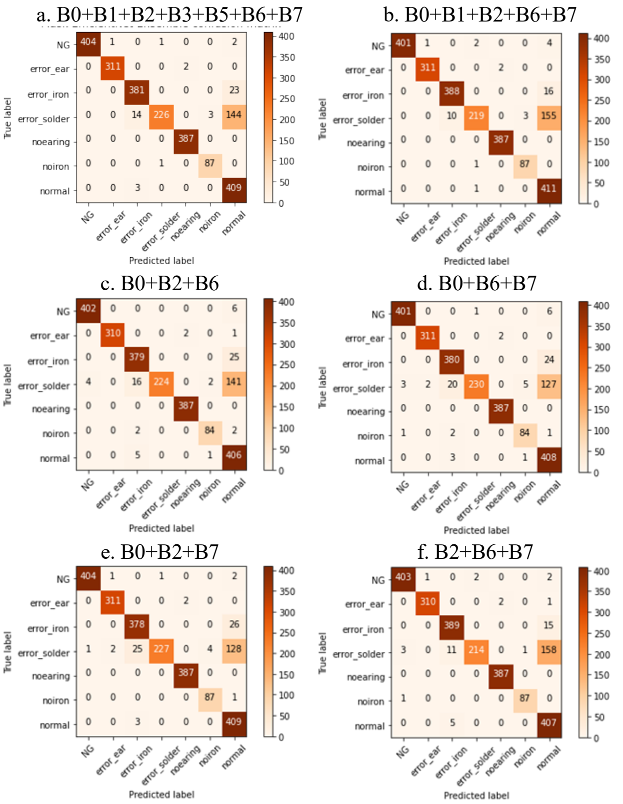
TABLE V. ENSEMBLE LEARNING TEST ACCURACIES

Ensemble Learning Combinations	Accuracy
B0+B1+B2+B3+B5+B6+B7	0.919132
B0+B1+B2+B6+B7	0.918716
B0+B2+B6	0.913714
B0+B6+B7	0.917465
B0+B2+B7	0.918299
B2+B6+B7	0.915798

In order to further investigate the potential for enhancing the accuracy of ensemble learning, this study performed weight adjustments on the ensemble models with only three constituent models. This choice of three models allows for better weight distribution based on individual model performance, determining the proportional weight for each model in the ensemble learning as shown in Table VI.

The weights in Table VI are assigned to models based on their better performances. Through various combinations and comparisons, the optimal weight distribution for each ensemble model is determined. From Table VI, it can be observed that after adjusting the weights, the overall accuracy also improves. The accuracy increases from the original 0.91 to 0.92 and even 0.93, with the highest reaching 0.93. The

Fig. 7. Confusion Matrix for Ensemble Learning Combinations in Table V.


a. B0+B1+B2+B3+B5+B6+B7

b. B0+B1+B2+B6+B7

c. B0+B2+B6

d. B0+B6+B7

e. B0+B2+B7

f. B2+B6+B7

average accuracy is improved to 92.5%, and the model achieves an error rate of only 7%, compared to the initial 2/3 error rate. There is also a significant increase in correct classifications for the "Error_Solder" category as shown in Figure 8.

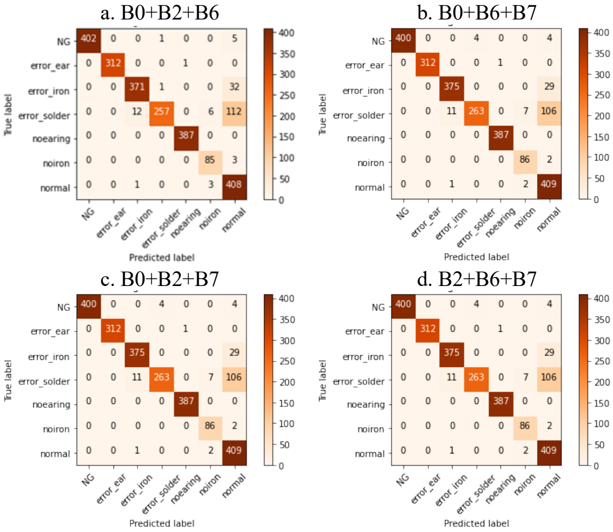
VII. Conclusion

Throughout the entire research process, it was observed that ensemble learning can indeed effectively improve the accuracy of object classification models, raising the accuracy from the original 0.89 to 0.91. Furthermore, when combined with adjusting the model's weights, ensemble learning further enhances the model's accuracy, increasing it to 0.92~0.93. However, during the research, it was found that regardless of using ensemble learning or adjusting the weights, there were still many misclassifications in the "Error_Solder" category. The model tended to classify them into the "Normal" category. This conclusion was drawn because the features of this category were not distinct enough, making it difficult for the machine to learn them. To achieve more accurate classification, the possible solutions are to increase the data volume or set new rules during data classification in the early stages, such as defining how much offset qualifies as an error in solder points. These are the key findings and conclusions of this research.

TABLE VI. Accuracy After Weight Adjustments

Ensemble Learning Combinations	Weight Distribution	Accuracy
B0+B2+B6	0.5+0.25+0.25	0.92621
B0+B6+B7	0.5+0.1+0.4	0.93038
B0+B2+B7	0.5+0.25+0.25	0.92997
B2+B6+B7	0.4+0.2+0.4	0.91663

Fig. 8. Confusion Matrix for Ensemble Learning Combinations in Table VI.


a. B0+B2+B6

b. B0+B6+B7

c. B0+B2+B7

d. B2+B6+B7

References

[1] E. O'Kelly, A. Arora, J. Ward, and P. J. Clarkson, "How Well do Face Masks Protect the Wearer Compared to Public Perceptions?," medRxiv, January, 2021, doi: 10.1101/2021.01.27.21250645.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature Vol. 521, Issue 7553, pp. 436-444, May 2015, doi: 10.1038/nature14539.

[3] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," arXiv:1905.11946, May, 2019 (v1), last revised, September, 2020 (v5), doi: 10.48550/arXiv.1905.11946.

[4] F. Huang, G. Xie, and R. Xiao, "Research on Ensemble Learning," 2009 International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, China, pp. 249-252, November, 2009, doi: 10.1109/AICI.2009.235.

[5] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," Journal of Big Data, Vol. 6, Article No. 60, pp. 1-48, July, 2019, doi: 10.1186/s40537-019-0197-0.

[6] K. Fukushima, "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," Biological Cybernetics, Vol. 36, pp. 193-202, April, 1980, doi: 10.1007/BF00344251

[7] K. Fukushima, "Neocognitron Capable of Incremental Learning," Neural Networks, Vol. 17, Issue 1, pp. 37-46, January, 2004, doi: 10.1016/S0893-6080(03)00078-9

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," in Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278-2324, November, 1998, doi: 10.1109/5.726791.

[9] K.-S. Oh and K. Jung, "GPU Implementation of Neural Networks," Pattern Recognition, Vol. 37, Issue 6, pp. 1311-1314, June, 2004, doi: 10.1016/j.patcog.2004.01.013.

[10] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a Convolutional Neural Network," 2017 International Conference on Engineering and Technology (ICET), pp. 1-6, Antalya, Turkey, August, 2017, doi: 10.1109/ICEngTechnol.2017.8308186.

[11] K. Wang, X. Liu, J. Zhao, H. Gao, and Z. Zhang, "Application Research of Ensemble Learning Frameworks," 2020 Chinese Automation Congress (CAC), pp. 5767-5772, Shanghai, China, November, 2020, doi: 10.1109/CAC51589.2020.9326882.