International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Improved Lightweight Deep Learning Model for the Fine-Feature Classification of Pipe Components

Changxing Chen[1], Afizan Azman[2]

Submitted: 03/12/2023 **Revised**: 27/01/2024 **Accepted**: 02/02/2024

Abstract: In the field of industrial manufacturing, the most common application of machine vision is object classification. Traditional models face challenges in classifying objects with small features and have yet to develop a complete universal algorithm, resulting in lower accuracy. To address these issues and enable the model's use on portable embedded devices, as well as facilitate the classification of pipe components with subtle internal features, a lightweight deep learning model has been developed. This model utilizes a relatively small dataset comprising specific types of work pieces occurring in actual factory production. The dataset involves the use of subtle image features for the classification of pipe components. The proposed model combines a fine-tuning module to capture multi-scale features of the input image and utilizes attention mechanisms to enhance the model's generalization ability. Not only enhances detection accuracy but also achieves network lightweight, with an accuracy of 96.33%. A comparison with other models demonstrates an improvement in accuracy of at least 4%, along with a significant reduction in both total and training parameters, meeting usability requirements, possessing lower computational complexity is essential to ensure fast and efficient operation in scenarios with embedded devices, mobile devices, or other resource-constrained environments.

Keywords: *Object classification; Subtle image features; Lightweight network; Industrial*

1. Introduction

In the field of industrial manufacturing, the stable, robust and accurate identification of workpieces is still a frontier topic and difficult problem [1, 2]. According to the actual industrial production, and provide an important classification model. This is very important. High-level features of the target can be learned from the noisy image, and this feature has good robustness to the shape change of the target to some extent. It provides an important reference for the design of the future workpiece classification model, and is of great significance to the intelligent manufacturing of industrial practical applications.

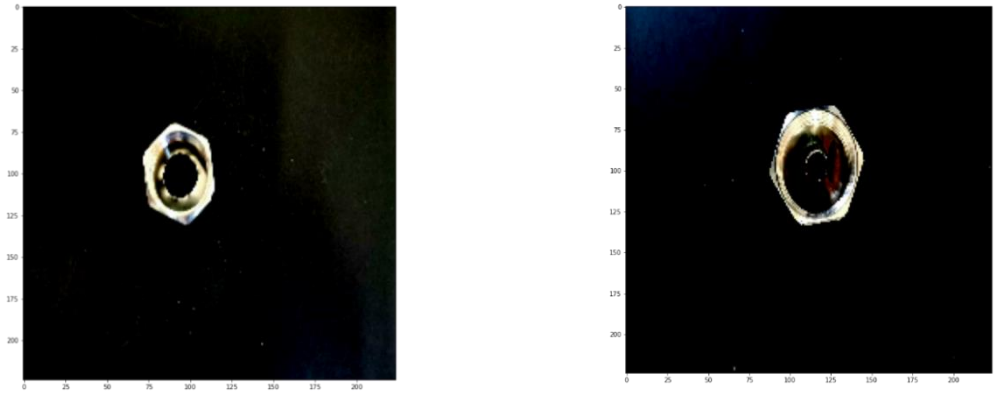
Because the traditional descriptors are the characteristics of manual design, when the shape of workpieces is complex and there are small differences between different workpieces, the traditional processing methods seem inadequate[3, 4]. The traditional method cannot well represent the feature difference between different workpieces with complex structure [5]. Therefore, how to detect and identify workpieces stably and efficiently is a problem and difficulty that has not been solved at present[6, 7]. At present, there are few workpieces in the field of industrial manufacturing. The recognition method based on deep neural network needs to manually mark a large number of datasets if it needs to use the datasets created by itself, this is also one of the restrictive factors that the current methods based on deep neural network cannot be well extended[8, 9]. The key is that there is no universal model that can recognize all images.

[1] *School of Computer Science and Engineering, Taylor's University, Subang Jaya,Malaysia.Email ID:1471280002@qq.com*
[2] *School of Computer Science and Engineering, Taylor's University, Subang Jaya,Malaysia.Email ID: Afizan.Azman@taylors.edu.my*

2. Data Collection and Processing

During the classification process, the workpieces share the same external shape, but their internal structures differ. The internal shape of the workpiece is divided into five types: small circle, big circle, square, pentagon, and hexagon. In the dataset, there are numerous images similar, The direct features of the data image are not very obvious,as shown in Figure 1. To efficiently handle these images, a data type conversion is applied to emphasize their data characteristics. This approach allows for quicker image preprocessing with the aim of retaining the original features of the images as much as possible [10, 11]. The goal is to speed up the model training process while maintaining the same recognition accuracy in deep learning models. The image samples are collected comprehensively, and the images inside the workpiece are gathered at different distances [12, 13]. For the same part, multiple shots are taken, and shooting conditions are not limited. It is ensured that the pictures taken each time cannot be repeated, and each picture is different. Therefore, the workpieces are classified based on their different internal shapes.The dataset described above is organized in a folder structure. Additionally, the images within the training data folder, validation data folder, and test data folder are randomly selected and distributed to their respective folders. This randomization process enhances model learning effectiveness and improves model stability [14-16]. Within the training data folder, each category contains 180 images, while the validation data folder has 60

images per category, and the test data folder contains 60 images per category. The entire dataset comprises 1800 images, all in JPG format.



(a) (b)

Fig 1. Comparison of different internal workpieces in processed dataset (a) big circle (b) small circle

Through the series of outlined image preprocessing steps, a dataset was created for the proposed model. This dataset consists of a total of 1500 images, categorized into five distinct classes, with each class containing 300 workpiece images. Following classification, the dataset is then subdivided into different sets: the training dataset comprises 900 images, the validation dataset consists of 300 images, and the test dataset includes 300 images.Before commencing training, the dataset is partitioned into training, validation, and test sets at a ratio of 3:1:1. The training process is specified to involve 75 iterations. In each iteration, a subset of the training dataset is randomly selected using the shuffle function, which shuffles the indices without affecting the correspondence between images and labels.

3. The Architecture of the Proposed Model

An optimization model was designed to increase the depth of the network and optimize the activation functions of neural networks and perceptrons. Firstly, it is essential to identify key information about the shape of the workpieces to be classified [17, 18]. Secondly, the classification of workpiece categories is achieved through labeled categorization, forming the training dataset. The completed training dataset serves as the foundation for classification detection [19, 20]. Before training and testing data, standardized preprocessing is applied to enhance data quality. Given that this is a small-sample dataset, training the model with limited samples requires adapting the model to the specific characteristics of the data,. In tasks involving the processing of image datasets, the initial layer is typically the input layer. The input images are matrices of processed pixels. The input layer can handle multidimensional data. While adhering to the MobileNetV2 network architecture, the goal is to ensure model generality and reduce model parameters. Input neurons connect to neurons in the next layer to perform convolutional layer operations, which are then stacked. The convolutional layer is the core of convolutional neural networks [21]. Each node in the convolutional layer receives input from only a small part of the neurons in the previous layer. Convolution deepens the matrix of the upper layer, and these steps are iterated to extract features from the images [22, 23]. The pooling layer can represent a region of pixels as a single pixel, discarding less important information to enhance the efficiency of the model.

Then, the input goes into the fine-tuning module, whose primary function is to adjust the pre-trained model through operations such as channel extraction, global average pooling, and reduction of feature map dimensions. Firstly, the channel extraction operation aids in extracting image features. Subsequently, the feature map dimensions are reduced through global average pooling, thereby decreasing the complexity of the model. Next, non-linearity is introduced through the ReLU activation function, allowing the model to better capture complex relationships within the data. Finally, the optimization algorithm of gradient descent is employed to fine-tune model parameters for adaptation to specific tasks, thereby enhancing model performance. The design of this module aims to make the model more adaptable to the new task's dataset, simultaneously speeding up the training process and reducing the risk of overfitting.

After the above operations,convolution operation is performed on input images, followed by batch normalization, max pooling, and activation. Then, a new backbone network structure is created by combining weight parameter update mechanisms and the SeNet(SE) channel attention mechanism. As shown in Figure 2， To enhance the effectiveness of image features, an attention mechanism has been incorporated into the network to improve the model's fitting ability by extracting relevant information

further. To improve the effectiveness of image features, SE attention mechanism is incorporated into the network to enhance the model's fitting capability by further extracting relevant information. SENet is a Squeeze-and-Excitation (SE) module for collecting global information, capturing inter-channel relationships, and improving representational capabilities.The Squeeze module collects global spatial information through Global Average Pooling.The Excitation module captures inter-channel relationships through the use of fully-connected and non-linear layers and outputs an attention vector. layer to capture the relationships between channels and output an attention vector. Each channel of the input feature is then scaled by multiplying it with the corresponding element of the attention vector. Equation is shown in (1),SE module F with X as input and Y as output, where W denotes the weight, σ denotes the sigmoid function, δ denotes the nonlinear activation function, and GAP denotes the Global Average Pooling operation.This mechanism allows the model to learn the importance of different local regions in the images.Using smaller convolutional kernels can reduce the number of parameters in each convolutional layer.

$$Y = sX = F_{se}(x,\theta)X = \sigma(W_2\delta(W_1 GAP(X)))X \quad (1)$$

The improved lightweight model structure, as illustrated in Figure 2, is characterized by the use of depth-wise separable convolutions instead of traditional convolutional operations. The channel size is gradually reduced by employing smaller convolutional kernels, effectively reducing the number of parameters. The model incorporates a fine-tuning module to capture multi-scale features of the input image. Adjustments are made through feature map blocks, maintaining a clear hierarchy and contributing to a more lightweight model. The traditional fully connected layers are replaced with global average pooling layers, significantly reducing the number of parameters. Adopting a dimensionality reduction strategy, adjustments to the attention mechanism aid the model in better adapting to diverse characteristics of input data, enhancing its generalization capability and overall adaptability. Replacing a 3x3 convolutional kernel with a 1x1 kernel or using depth-wise separable convolutions. Replacing traditional fully connected layers with global average pooling layers can significantly decrease the number of parameters. Global average pooling transforms the entire feature map into a scalar, serving as a substitute for the parameters in fully connected layers. Applying these methods to the model can significantly reduce the number of parameters.

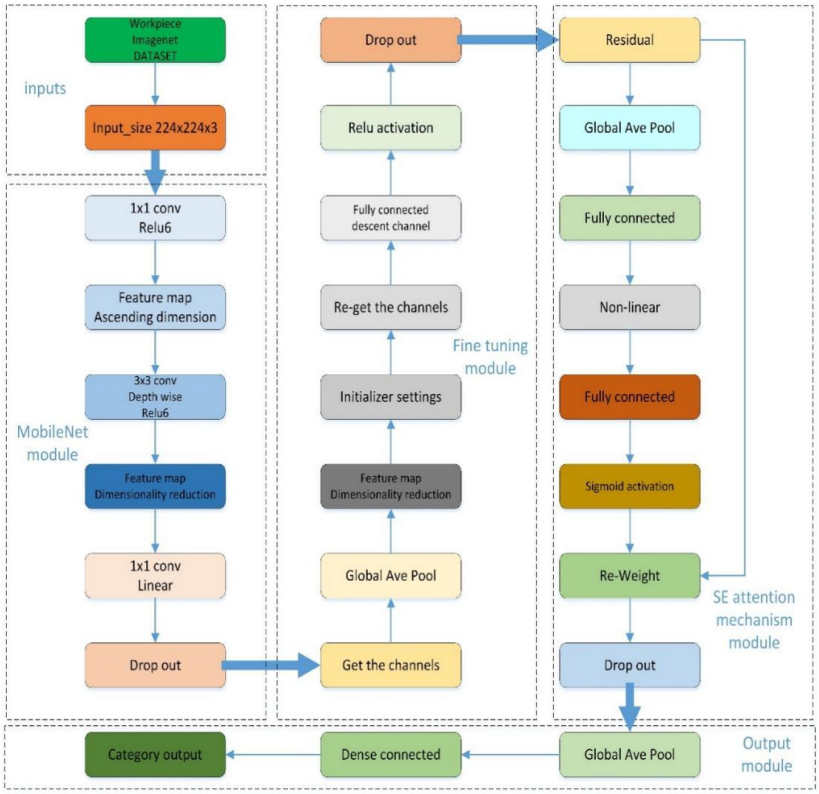


Fig 2. Proposal lightweight model structure

Learning networks can effectively address training issues with small-sample datasets. Before commencing training in the neural network, it is necessary to fine-tune the learning rate and configure the number of training samples per batch. The model's ability to extract meaningful features from images is a crucial factor in achieving better classification results. During the training process, progress is monitored by checking if the iteration limit has been reached. If the limit is not reached, training continues, and weight parameters are updated. This process involves observing the loss function and accuracy parameters to determine if they have reached their maximum values. Finally, when the

iteration limit is reached, training is stopped, and the trained neural network model is output.To further analyze the performance and effectiveness of the trained model, it is crucial to evaluate its performance on the test dataset for small-sample image classification.

This mechanism allows the model to learn the importance of different local regions in the image. During the training process, progress is monitored by checking whether the iteration limit has been reached. If the limit has not been reached, training continues, and the weight parameters are updated. This process involves observing the loss function and accuracy parameters to determine if they have reached their maximum values. Finally, when the iteration limit is reached, training is halted, and the trained neural network model is outputted. To further analyze the performance and effectiveness of the trained model, it is crucial to evaluate its performance on a test dataset for small-sample image classification. The effectiveness of image features can be understood as the model's ability to represent images effectively, thereby enhancing classification performance. This mechanism allows the model to learn the importance of different local regions in the image. During the training process, progress is monitored by checking whether the iteration limit has been reached. If it has not been reached, training continues, and the weight parameters are updated. This process involves observing the loss function and accuracy parameters to determine if they have reached their maximum values. Finally, when the iteration limit is reached, training is stopped, and the trained neural network model is outputted.

The model output process is illustrated in Figure 3.， Perform a single convolution operation, followed by batch normalization, max-pooling, and activation on the input image. Then, incorporate a weight parameter updating mechanism and the SeNet channel attention mechanism to create a novel backbone network structure. While adhering to the MobileNetV2 network architecture, ensure both model generalization and a reduction in model parameters. This mechanism allows the model to learn the importance of different local regions in the image. During training, the progress is monitored by checking whether the iteration limit has been reached. If the limit has not been met, training continues, and weight parameters are updated. The process involves observing the loss function and accuracy parameters to determine if they have reached their maximum values. Finally, when the iteration limit is reached, training is halted, and the trained neural network model is output.

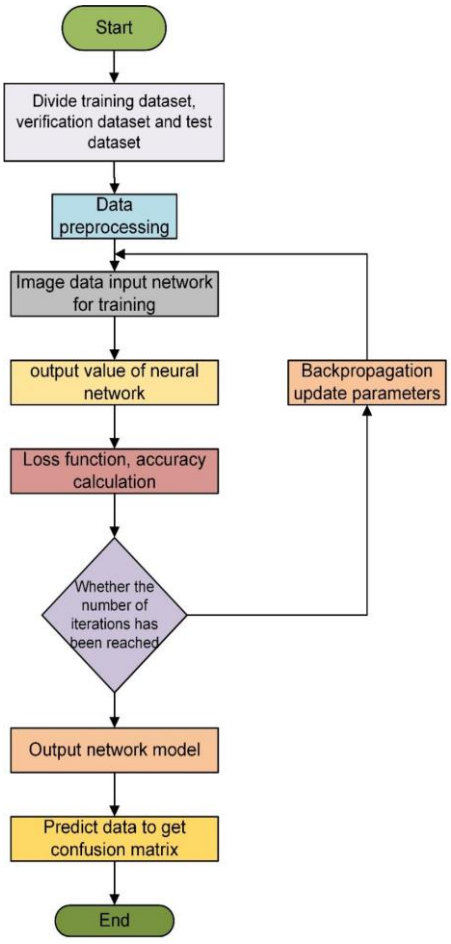
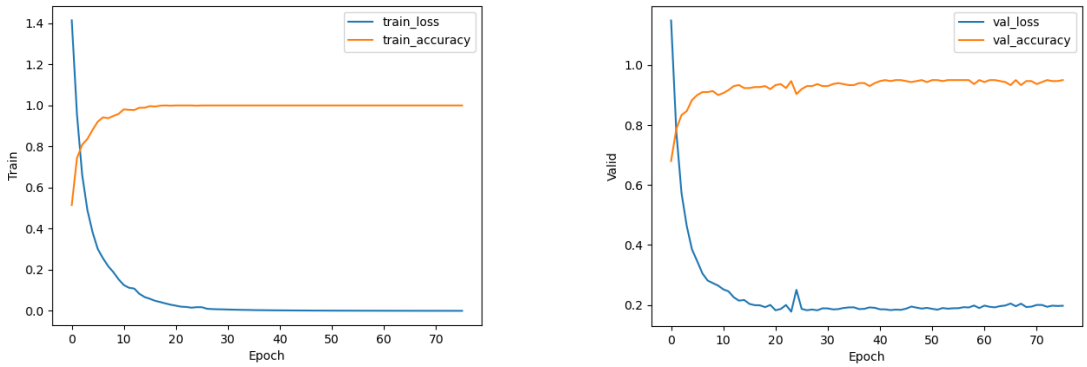


Fig 3. Model Experimental Flow Chart

The process involves pre-processing a target set into a suitable array, training the MobileNetV2 with ImageNet weights, and implementing a customized network refinement. This refinement includes freezing specific layers, applying channel attention mechanisms, and utilizing fully connected layers with careful initialization and activation functions. The training process incorporates optimization and epoch checks, leading to a final step of global average pooling and a fully connected layer for classification output.1.Model Training:Train the MobileNet network structure.Load pre-trained weights trained on ImageNet.Freeze the weights of the feature extraction layer to ensure they remain fixed during training. 2. Fine-tuning Module: Through training on a specific task, this module adjusts model parameters to accommodate the task's specific features, thereby enhancing the model's performance on that particular task.3.Network Modification:Implement global average pooling.Downsize the number of channels in the feature map.Apply initialization for normalization.Utilize the Senet channel attention mechanism 。 4. Global average pooling, compression of the feature map into a feature vector, resizing channels, applying activation functions , and generating weighted feature maps.

4. Results and Discussion

In classification research, accuracy serves as a metric for evaluating both training and prediction performance. The network's learning rate during training on the training and validation sets is set at 0.0001, with a batch size of 32 and 75 epochs. The input dataset images are divided into training, testing, and validation sets in a 3:1:1 ratio. Notably, the network quickly achieves stability in accuracy on the training dataset. As shown in Figure 4(a)， Following training, testing and validation accuracies exhibit gradual improvement with additional iterations. As shown in Figure 4(b) presents experimental results where the model is trained on the validation set, and test set accuracy serves as the evaluation metric. It's clear that with an increase in the total number of model parameters and network layers, the model's accuracy on the validation set steadily improves. Concurrently, accuracy reaches a peak of 0.95, sustaining at that level after 75 iterations. Deeper networks exhibit smaller oscillations in the loss function, and training accuracy improves more rapidly, indicating enhanced fitting capabilities. This confirms that the model meets convergence criteria and effectively accomplishes the recognition task.



(a) (b)

Fig 4. The performance of the C-Lightweight mode (a) on the training dataset (b) on the validation dataset

The confusion matrix score results in the train data set for the proposed model are shown in Figure 5. From the number of samples on the main diagonal of the confusion matrix, it is evident that the lightweight model proposed in this research achieved 100% recognition accuracy for each category on the training dataset. This demonstrates the model's ability to effectively distinguish between different categories. Furthermore, the test results in the test set are shown in Figure 6. The model's overall performance on the test dataset, as calculated from the confusion matrix, indicates an accuracy of 96.33%. This suggests that the model is capable of accurate classification, and the improvement in accuracy opens up the possibility for large-scale individual detection. Additionally, the model stabilizes after 75 iterations. An issue arises where accuracy for classes A and D is lower compared to other classes. This could be related to the challenge of capturing features for these two classes in 2D images and their similarity in classification features, making it difficult to effectively differentiate between them. While accuracy in recognition is somewhat reduced due to features being more easily confused with other categories in limited image resolution, when features are distinctive, there is a significant improvement in the metrics. Overall accuracy has been notably enhanced.

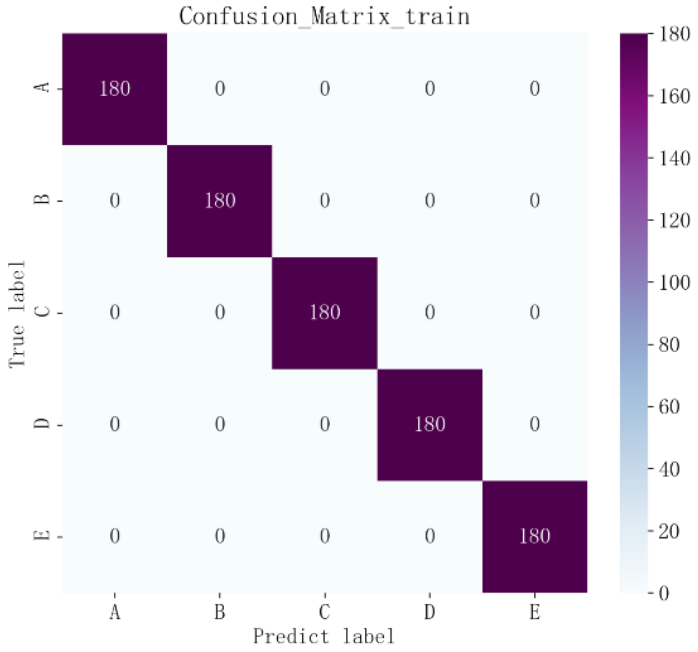


Fig 5. The performance of the Proposal lightweight model in train data set

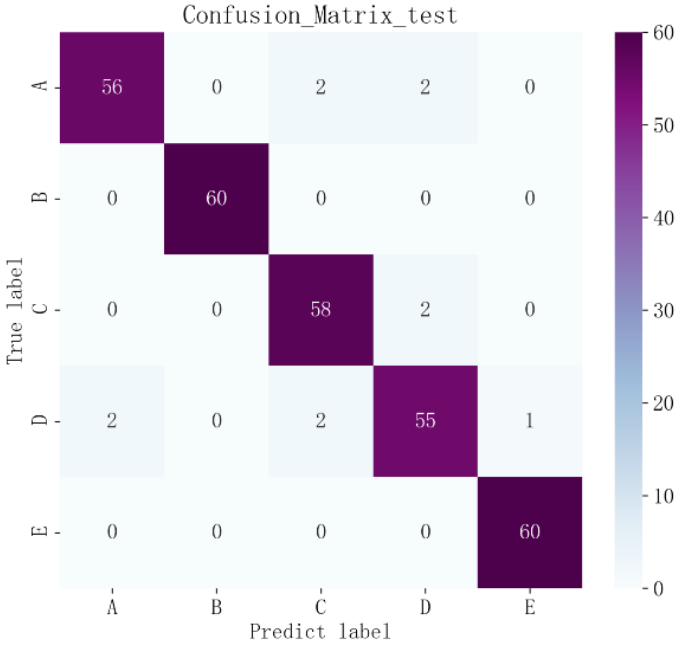


Fig 6. The performance of the Proposal lightweight model in test dataset

To analyze the impact of incorporating attention mechanisms into the proposed lightweight model, experiments were conducted using different attention mechanisms on the same test dataset. The improved model, with the addition of SENet, CBAM, ECA, SAM, and GAM attention mechanisms at the same positions, underwent experiments for comparison. Prior to entering the attention mechanism, the data passed through the Fine Tuning Module of the model. The fine-tuning module enables the model to adapt to specific features of the new task, thereby enhancing the model's performance in specific domains or applications. To ensure a fair evaluation of the impact of each attention mechanism, the fine-tuned model was included when testing the attention mechanisms. The detection results are shown in Table 1. Experiment A, which introduced the SE attention mechanism, achieved the highest detection accuracy while maintaining competitive detection speed compared to other experiments. It is noteworthy that Experiment F, despite incorporating the SE attention mechanism, did not include the Fine Tuning Module, resulting in lower accuracy and only marginal improvement in detection speed. This indicates that simply adding attention mechanisms to the model may not effectively suppress background interference. Furthermore,

it highlights that the proposed network model's fine-tuning module can efficiently extract key features of various objects. The experimental results demonstrate that the fine-tuning module proposed in this paper, along with the SE attention mechanism, can achieve feature perception, enhance feature extraction in regions of interest, effectively strengthen feature representation, and further improve the model's robustness.

Table 1. The performance results of the attention mechanism model

Experimental Group	Fine tuning module	SENet	CBAM	ECA	SAM	GAM	Accuracy
Experiment A	√	√	x	x	x	x	0.9633
Experiment B	√	x	√	x	x	x	0.9483
Experiment C	√	x	x	√	x	x	0.9287
Experiment D	√	x	x	x	√	x	0.9048
Experiment E	√		x	x	x	√	0.9351
Experiment F	x	√	x	x	x	x	0.9162

The performance of the Proposal lightweight model was evaluated through dataset classification. Deep learning experiments were conducted, comparing the Proposal lightweight model with classical networks such as MobileNet V2, VGG16, ResNet50, Ghostnet, and SqueezeNet. The comparative method involved training all the models using the same training and validation sets and evaluating their performance on the same test set.The models' parameters were optimized using the Adam gradient descent optimization method, with an initial learning rate set to 0.0001. Only the final layer of each model was modified to output the five classes. The results of the model comparison experiment are presented in Figure 7. It can be observed that, when compared to lightweight models such as MobileNet V2, Ghostnet, and EfficientNet, the Proposal lightweight model outperforms them in terms of accuracy. The traditional lightweight models mentioned typically exhibit accuracy rates generally below 90% on both the validation and test sets. In comparison to classical models like VGG16 and ResNet50, the Proposal lightweight model shows higher accuracy, with the ResNet50 model achieving the highest accuracy at 92%. The accuracy of The proposed lightweight model can reach 96.33% which At least 4% higher than the test model.This suggests that the improved Proposal lightweight model demonstrates the highest robustness, best generalization, highest accuracy, and the highest practical value among the models compared.

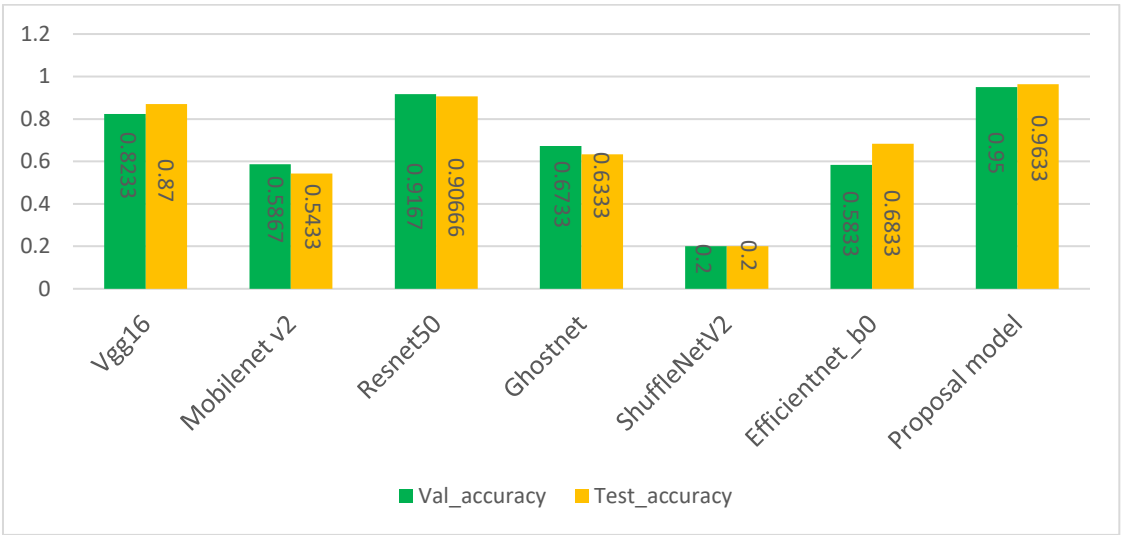


Fig 7. Comparison of different model in accuracy

To further affirm the lightweight nature of the proposed model, this experiment compares the parameters and computational speed of the Proposal lightweight model with the aforementioned models. For a more detailed understanding of the model's parameters, a dataset from the same database, tailored for an identical classification task, was initially selected. After training the weight parameters, the test accuracy of the experimental neural models was compared. As shown in Table 2, the experimental results illustrate the total model parameters and the trainable

parameters required for training across all models on this dataset. The results indicate that under the same dataset conditions and within the same testing environment, the Proposal lightweight model achieved an average recognition rate of 96.33% with a relatively lower number of parameters. It outperformed convolutional models (Vgg16 and Resnet50) by an average of 4% to 13%. In comparison to their structures, the total number of parameters and trainable parameters were reduced by approximately 88% to 98%.

The Proposal lightweight model demonstrated a substantial 40% to 70% improvement in the average recognition rate compared to relatively lightweight convolutional models like MobileNet V2 and Ghostnet. As shown in Table 1, interestingly, when considering its structure, the increase in the total parameter count was only approximately 4%. In contrast, the Proposal lightweight model utilized only 50% to 78% of the parameters necessary for model training. When compared to the ShuffleNetV2 model, although the Proposal lightweight model had slightly more than twice the total parameter count, the parameters used for training accounted for only 52% of those required by the ShuffleNetV2 model. This noteworthy reduction in parameter usage resulted in an impressive 79% improvement in accuracy. Clearly, the Proposal lightweight model has managed to achieve excellent predictive performance while significantly reducing the parameter count.In comparison to the EfficientNet model, not only did the total parameters and trainable parameters decrease substantially, but the accuracy also improved by an impressive 29% to 38%. As a result, the designed model demonstrates a significant advantage in recognition accuracy over both traditional and lightweight models. However, in comparison to both of these, the Proposal lightweight model proposed in this paper achieves a substantial reduction in parameter count.

Table 2. Table of parameter quantities for different models

Model	Total params	Trainable params
Vgg16	134281029	134281029
Mobilenet v2	2279941	2245829
Resnet50	23571397	23518277
Ghostnet	2695253	2673285
ShuffleNetV2	1196565	1183361
Efficientnet_b0	4055969	4013953
Proposal model	2823509	565525

To further validate the algorithm's lightweight performance in executing classification tasks, GFLOPs comparison experiments were conducted between the Proposal lightweight model and other models. GFLOPs, understood as the computational workload of the model, provide a metric for assessing the model's complexity. The experimental results, as shown in Figure 8, reveal that the Proposal lightweight model has a floating-point operation count of only 7.58E-05, which is significantly lower than the GFLOPs of other models. This indicates that the proposed Proposal lightweight model is constructed in a more scientifically efficient manner, offering enhanced generalization and real-time capabilities. It better fulfills the practical requirements of portable detection needs.

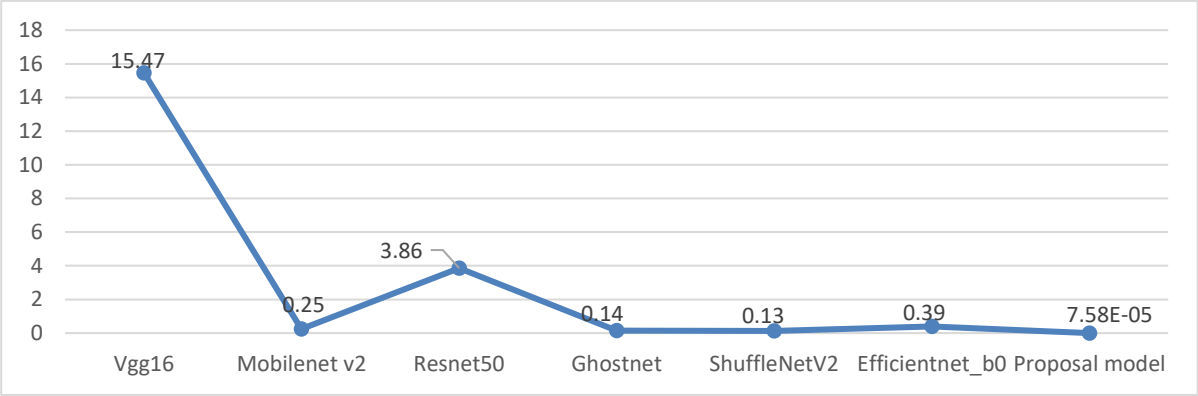


Fig 8. Comparison of operational performance of different models（GFLOPs）

5.

6. Conclusion

The proposed model integrates a fine-tuning module to capture multi-scale features of the input image and utilizes an attention mechanism to enhance the model's generalization capability which can enhance the model's robustness and generalization capabilities. The proposed method involves designing a lightweight module for small target detection, enabling efficient detection of small targets with the fewest possible parameters. The proposed model attains an accuracy of 96.33%, surpassing the test model by at least 4%. This highlights the enhanced robustness, superior generalization, highest accuracy, and practical value compared to other models. Additionally, the proposed model excels in predictive performance while significantly reducing parameter count.This approach addresses the issue of low detection accuracy in traditional lightweight models when dealing with insufficient small sample data. It resolves the challenge of effectively deploying models in environments with limited computational resources and low detection rates that prevent real-time detection.

References

[1] S. G. Karnaukh, O. E. Markov, A. A. Shapoval, and N. S. Hrudkina, "Selecting a cutting method for workpieces before stamping using synergetic fracture criteria and a deformability limit determination technique for separating processes," International Journal of Advanced Manufacturing Technology, Article vol. 129, no. 11-12, pp. 5447-5455, 2023, doi: 10.1007/s00170-023-12627-z.

[2] G. Zhou et al., "A new algorithm for chatter quantification and milling instability classification based on surface analysis," Mechanical Systems and Signal Processing, Article vol. 204, 2023, Art no. 110816, doi: 10.1016/j.ymssp.2023.110816.

[3] F. Khan, K. Kamal, T. A. H. Ratlamwala, M. Alkahtani, M. Almatani, and S. Mathavan, "Tool Health Classification in Metallic Milling Process Using Acoustic Emission and Long Short-Term Memory Networks: A Deep Learning Approach," IEEE Access, Article vol. 11, pp. 126611-126633, 2023, doi: 10.1109/ACCESS.2023.3328582.

[4] P. Zhang et al., "Improving generalisation and accuracy of on-line milling chatter detection via a novel hybrid deep convolutional neural network," Mechanical Systems and Signal Processing, Article vol. 193, 2023, Art no. 110241, doi: 10.1016/j.ymssp.2023.110241.

[5] S. Zhu, G. Fu, Y. Zheng, Z. Li, and J. Yang, "Universal Surface Texture Modeling Method for Five-axis Surface Milling," Zhongguo Jixie Gongcheng/China Mechanical Engineering, Article vol. 34, no. 16, pp. 1946-1957, 2023, doi: 10.3969/j.issn.1004-132X.2023.16.008.

[6] A. Fertig, M. Weigold, and Y. Chen, "Machine Learning based quality prediction for milling processes using internal machine tool data," Advances in Industrial and Manufacturing Engineering, Article vol. 4, 2022, Art no. 100074, doi: 10.1016/j.aime.2022.100074.

[7] D. A. Molitor, C. Kubik, R. H. Hetfleisch, and P. Groche, "Workpiece image-based tool wear classification in blanking processes using deep convolutional neural networks," Production Engineering, Article vol. 16, no. 4, pp. 481-492, 2022, doi: 10.1007/s11740-022-01113-2.

[8] C. Chen, A. Abdullah, S. H. Kok, and D. T. K. Tien, "Review of Industry Workpiece Classification and Defect Detection using Deep Learning," International Journal of Advanced Computer Science and Applications, Article vol. 13, no. 4, pp. 329-340, 2022, doi: 10.14569/IJACSA.2022.0130439.

[9] Q. Li, Z. Luo, H. Chen, and C. Li, "An Overview of Deeply Optimized Convolutional Neural Networks and Research in Surface Defect Classification of Workpieces," IEEE Access, Article vol. 10, pp. 26443-26462, 2022, doi: 10.1109/ACCESS.2022.3157293.

[10] C. H. Wang et al., "Deep Learning-based Diagnosis and Localization of Pneumothorax on Portable Supine Chest X-ray in Intensive and Emergency Medicine: A Retrospective Study," Journal of medical systems, Article vol. 48, no. 1, p. 1, 2023, doi: 10.1007/s10916-023-02023-1.

[11] D. Wu, J. Yang, M. U. Ahsan, and K. Wang, "Classification of integers based on residue classes via modern deep learning algorithms," Patterns, Article vol. 4, no. 12, 2023, Art no. 100860, doi: 10.1016/j.patter.2023.100860.

[12] Y. Dou, J. Xia, M. Fu, Y. Cai, X. Meng, and Y. Zhan, "Identification of epileptic networks with graph convolutional network incorporating oscillatory activities and evoked synaptic responses," NeuroImage, Article vol. 284, 2023, Art no. 120439, doi: 10.1016/j.neuroimage.2023.120439.

[13] E. Tjoa, H. J. Khok, T. Chouhan, and C. Guan, "Enhancing the confidence of deep learning classifiers via interpretable saliency maps," Neurocomputing, Article vol. 562, 2023, Art no. 126825, doi: 10.1016/j.neucom.2023.126825.

[14] Z. H. Ren et al., "DeepMPF: deep learning framework for predicting drug–target interactions based on multi-modal representation with meta-path semantic analysis," Journal of Translational Medicine, Article

vol. 21, no. 1, 2023, Art no. 48, doi: 10.1186/s12967-023-03876-3.

[15] D. Tabernik, M. Šuc, and D. Skočaj, "Automated detection and segmentation of cracks in concrete surfaces using joined segmentation and classification deep neural network," Construction and Building Materials, Article vol. 408, 2023, Art no. 133582, doi: 10.1016/j.conbuildmat.2023.133582.

[16] X. Xiahou, Z. Li, J. Xia, Z. Zhou, and Q. Li, "A Feature-Level Fusion-Based Multimodal Analysis of Recognition and Classification of Awkward Working Postures in Construction," Journal of Construction Engineering and Management, Article vol. 149, no. 12, 2023, Art no. 04023138, doi: 10.1061/JCEMD4.COENG-13795.

[17] A. N. Arun, P. Maheswaravenkatesh, and T. Jayasankar, "Facial Micro Emotion Detection and Classification Using Swarm Intelligence based Modified Convolutional Network," Expert Systems with Applications, Article vol. 233, 2023, Art no. 120947, doi: 10.1016/j.eswa.2023.120947.

[18] Y. Cai, Z. Yao, X. Cheng, Y. He, S. Li, and J. Pan, "Deep metric learning framework combined with Gramian angular difference field image generation for Raman spectra classification based on a handheld Raman spectrometer," Spectrochimica Acta - Part A: Molecular and Biomolecular Spectroscopy, Article vol. 303, 2023, Art no. 123085, doi: 10.1016/j.saa.2023.123085.

[19] O. F. Abd-Elaziz, M. Abdalla, and R. A. Elsayed, "Deep Learning-Based Automatic Modulation Classification Using Robust CNN Architecture for Cognitive Radio Networks," Sensors, Article vol. 23, no. 23, 2023, Art no. 9467, doi: 10.3390/s23239467.

[20] R. Yousaf et al., "Satellite Imagery-Based Cloud Classification Using Deep Learning," Remote Sensing, Article vol. 15, no. 23, 2023, Art no. 5597, doi: 10.3390/rs15235597.

[21] T. L. Huang et al., "Transfer learning with CNNs for efficient prostate cancer and BPH detection in transrectal ultrasound images," Scientific Reports, Article vol. 13, no. 1, 2023, Art no. 21849, doi: 10.1038/s41598-023-49159-1.

[22] Y. Chai, H. Yu, L. Xu, D. Li, and Y. Chen, "Deep Learning Algorithms for Sonar Imagery Analysis and Its Application in Aquaculture: A Review," IEEE Sensors Journal, Article vol. 23, no. 23, pp. 28549-28563, 2023, doi: 10.1109/JSEN.2023.3324438.

[23] H. Liu et al., "A Deep Learning Neural Network Method Using Linear Eigenvalue Statistics for Schizophrenic EEG Data Classification," Mathematics, Article vol. 11, no. 23, 2023, Art no. 4776, doi: 10.3390/math11234776.