IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 2, 2022, accepted April 11, 2022, date of publication April 20, 2022, date of current version April 27, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3168861

LPViT: A Transformer Based Model for PCB Image Classification and Defect Detection

KANG AN[1] AND YANPING ZHANG[2]
[1]Qianjiang College, Hangzhou Normal University, Hangzhou 311121, China
[2]Department of Computer Science, Gonzaga University, Spokane, WA 99258, USA

Corresponding author: Kang An (q0070031@huqc.edu.cn)

This work was supported by the Hangzhou Normal University Science Foundation under Grant 2022QJJL08.

ABSTRACT PCB (printed circuit board) is an extremely important component of all electronic products, which has greatly facilitated human life. Meanwhile, tons of PCBs in the waste streams become a waste of resources, which puts the recycling and reuse of PCBs in urgent need. In the manufacturing and recycling of electronic products, the classification of PCBs, recognition of sub-components, and defect detection have been the key technology. Traditional manual detection and classification are subjective and rely on individuals' experience. With the development of artificial intelligence, lots of research efforts have been dedicated to the automated detection and recognition of PCBs. In this paper, we propose a transformer-based model, LPViT, for defect detection and classification of PCBs. We conduct the defect detection task on the dataset DeepPCB, which consists of six different types of PCB defects. Defect detection benefits both manufacturing and recycling of PCBs. Among many electronic products, a group of affordable, general-purpose, and small-size PCBs is very popular, which are referred to as micro-PCBs. The classification and recognition of those PCBs will greatly facilitate the recycling and reuse process. We conduct the classification task on a dataset called **micro-PCB**, which includes 12 types of popular, general-purpose, affordable, and small PCBs. Through comparative experiments, our system demonstrates its advantage in both classification and defect detection tasks.

INDEX TERMS Classification, defect detection, label smooth, micro-PCB, DeepPCB, transformer, mask patch prediction, recognition.

I. INTRODUCTION

In 1925, Charles Ducas of the United States printed out the line pattern on an insulated substrate and then plated it to build up the wires, which started a new era of the modern PCB (printed circuit board) technology. At the end of the 20th century, Rigid-Flex, buried resistance, buried capacitance, metal substrates, and other new technologies continued to emerge. Right now, PCB has become an extremely important component of all electronic products and plays a pivotal role in electronic industry. The electronic industry is driven by Moore's Law. Products are becoming more and more powerful and integrated. The signal rate is getting faster and the product development cycle is getting shorter. As electronic products continue to miniaturize, PCB design faces various challenges brought by high speed and high density.

On the other hand, with the increasing usage of electronic devices, more and more PCBs become wastes in the waste streams. Many of them are functional or contain functional components (e.x., tantalum capacitors and ceramic filters) that are expensive and long-lasting. They also contain valuable chemical elements (rare earth elements, gallium, etc) necessary for producing electronic products. The recent auto crisis during COVID-19 caused by the shortage of chips and small electronic goods makes those resources even more precious. Therefore recycling and reuse of PCBs are getting more attention.

During the manufacturing, recycling, and reuse of electronic products, the classification of PCBs, recognition of electronic components on PCBs, and defect detection have been the key technology. In practice, there are a massive number of different PCB makes and models. With the development of machine learning and deep learning technologies, there has been plenty of research on classification and recognition in the field of computer vision using intelligent models.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhouyang Ren.

The classification and recognition of PCBs have some similarities with the problems in computer vision. However, there are much more diversities in PCBs. For example, compared to the human face, PCBs have electronic components (like capacitors, resistors, and integrated circuits) organized following various patterns. Even on a small PCB, there are many more features and those features are quite similar. Especially, some modern capacitors and resistors are small surface-mounted, which almost look the same while eyes, noses, and mouths are relatively easier to identify.

These years, lots of research efforts have been dedicated to the automated classification and recognition of components on PCBs while most of the research is more focused on classical datasets such as ImageNet [1]. However, as explained in [2], only a few PCB makes and models could be far more common. A typical candidate for these ''far more common'' PCBs would be general-purpose, affordable, and small PCBs, which are referred to as micro-PCBs [2]. The dataset we use in the classification task is called **micro-PCB** [2]. If the make and model of a PCB could be identified, we could conveniently identify the PCBs that we want to reuse. The sub-components used in it could also be directly recognized through a bill of materials. Meanwhile, defect detection has been necessary in real manufacturing and recycling process. It's challenging to conduct manual detection. To improve efficiency, it is necessary to develop an automated defect detection system.

In this research, we propose a new transformer-based model called LPViT and its corresponding training strategy to achieve accurate classification and defect detection on PCBs, and improve the robustness of the model. Our model can be used for both defect detection and classification tasks. We conduct classification tasks on the micro-PCB [2] dataset and the defect detection task on DeepPCB [3] dataset. The experimental results demonstrate the advantage of our model, which is the current SOTA model.

The main contributions of this research are as follows:

- Introducing the transformer structure to the classification task of PCB image data;
- Using a label to improve the robustness of the model smooth strategy;
- Driving the model to learn the relationship between different patches by mask patch prediction to ensure its extraction of deep relationships.

The rest of this paper is organized as follows. Section II reviews related research work. Section III explains our proposed method. In Section IV, comprehensive experiments are conducted to evaluate the effectiveness of the proposed method. Finally, in Section V, we conclude the paper.

II. RELATED WORKS

A. DETECTION AND CLASSIFICATION OF PCBS

The defect detection and classification of PCBs are critical for modern electronic circuits, which greatly impact the precision of circuit performance. Image processing has been applied for PCB defect detection, classification, and localization, which counters difficulties and subjective aspects in manual inspection and provides fast assessments. Malge and Nadaf [4] proposed a PCB defect detection and classification system to detect and classify defects in order to identify root causes. The system was based on a morphological image segmentation algorithm and image processing theories. The system focused on images of single-layer PCBs. Kamalpreet [5] proposed an algorithm to classify 14 defects into five groups, which used MATLAB image processing operations. With the development of machine learning and deep learning, more effective models have been developed. In [6], a deep learning-based advanced PCB inspection system was proposed. The classification task of PCBs is often limited by a limited dataset. The research applied image augmentation to get better performance. Ankit [7] created a deep learning model for Image Classification for PCBs, which aimed to detect defective PCBs and classify them as Good or Bad. In [8], to extract PCB circuits, machine learning and computer vision methods were investigated, designed, and tested with real-world PCB data. Deep learning networks (Faster R-CNN) and unsupervised machine learning clustering (XOR-based Kmeans) were used to implement the detection and localization of electronic components. However, all the above research aims to classify PCBs, identify electronic components, and detect defects that could be harmful to the circuit performance. Massive patterns of electronic components, their combinations and locations on PCBs, and limitations from the PCB image dataset make the classification and recognition way more complicated and less effective. In this research we are classifying micro PCBs based on their makes and models,which will greatly facilitate downstream tasks by conveniently retrieving sub-components through a bill of materials.

Meanwhile, computer vision has made significant progress. Its application in object detection [9], [10] has advanced the development of autonomous vehicles [11], robotics [12], and many other practical applications. There is a potential to adapt those implementations to other detection and classification tasks, such as PCB images. However, large-scale datasets are required to train those object detection networks to obtain good performance. Unfortunately, for the PCB tasks, it is infeasible to build up a large-scale dataset to train those networks.

B. TRANSFORMER MODELS

The transformer [13] mechanism was originally utilized in NLP (Natural Language Processing). Being applied in extended fields, it has demonstrated excellent performance in different applications. The evolvement of transformer models includes Bert [14], ALBERT [15], and RoBERTa [16]. Leading technology companies, like Google AI Lab, Amazon AI Lab, Baidu's ERNIE [17] and ERNIE2 [18] have been driven to devote resources to the research.

Some researchers have attempted to migrate the transformer's processing of input plenary data for use in the field of computer vision with great success, demonstrating

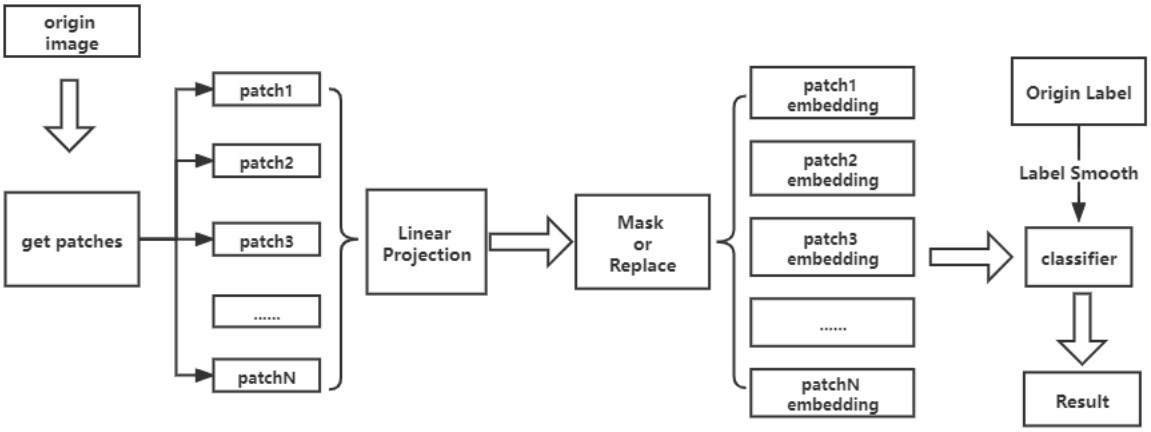

FIGURE 1. Overview of LPViT.

the general applicability and design rigor of such structures. In the field of computer vision, many transformer-based architectures have been presented, each of which is tailored to the specific task. For example, DETR [19] was created for object detection, and most AI researchers choose SETR [20] for semantic segmentation.

For classification tasks, ViT [21] and DeiT [22] have been proposed, which improve the metrics and become the basis for complex tasks such as detection and segmentation. Transformer-based strategies can also be well adapted to other applications. Taking the medical image classification [23], [24] as an example, transformer-based models can be adapted for tumor detection [25] and semantic segmentation.

However, for PCB classification, there is still no specific classification model. The PCB image has a variety of colors and components. For accurate classification, the features of PCB images need to be studied and accurately modeled.

III. METHOD

The capacitors, resistors, and other components on micro PCBs can vary significantly under different illumination and viewing angles due to the reflection of light. The smaller scale compared to ordinary PCBs further impacts on the results of classifications, which requires our proposed model to be extremely robust to eliminate the above interference.

In this research, we propose a new model, Label Robust and Patch Correlation Enhanced ViT (LPViT), which ensures the robustness of the model while fully exploiting the relationship between different regions of PCB images. Figure 1 shows the architecture of our proposed model, LPViT. In our model, the input image is first chunked to get different patches, and some of them are randomly masked or replaced, which aims to improve the ability of mutual understanding between different regions of the image. The operation is designed to drive the model to recover the lost information, and further extract features to get the patch embedding. With the label smooth strategy for training to improve the robustness, our model is able to achieve better results.

A. TRANSFORMER

First proposed by the Google team in 2017, Transformer is an architecture that has demonstrated its effectiveness in natural language processing. Completely abandoning many of the previous approaches, its innovations allow for significant improvement of the metrics of several natural language processing tasks at the time and laid the groundwork for the subsequent models. Figure 2 shows the overall architecture of the Transformer.

1) ATTENTION MECHANISM

It is widely believed that it is the Attention mechanism that contributes to the success of the Transformer architecture, which enhances the feature extraction capability. Attention is a mechanism for improving the effectiveness of RNN[15] (LSTM or GRU)-based Encoder + Decoder models, commonly known as Attention Mechanism.

For example, in machine translation and speech recognition, every single word will be assigned a weight, which makes the learning of neural networks more flexible (soft). Attention can be utilized to explain what exactly the model has learned, as well as the alignment relationship between the input and output sentences of translations.

In a transformer, the most important part is the Attention mechanism that is specially designed to promote the feature extraction function of the model for a better global and contextual understanding of the input information. To measure the relative importance of input features, the cosine similarity, also known as dot-product, is adopted. We use Dot-product Attention to describe the calculation of attention weights of different features.

A similar operation is used at all time steps, here as we get input key noted as $k_i, i = 1 \ldots N$, the corresponding values $v_i, i = 1 \ldots N$ and the query that we search, noted as q, we get formula 1 as follows:

$$Attention(Q, K, V) = \sum_{i=1}^{N} q^T k_i v_i \tag{1}$$

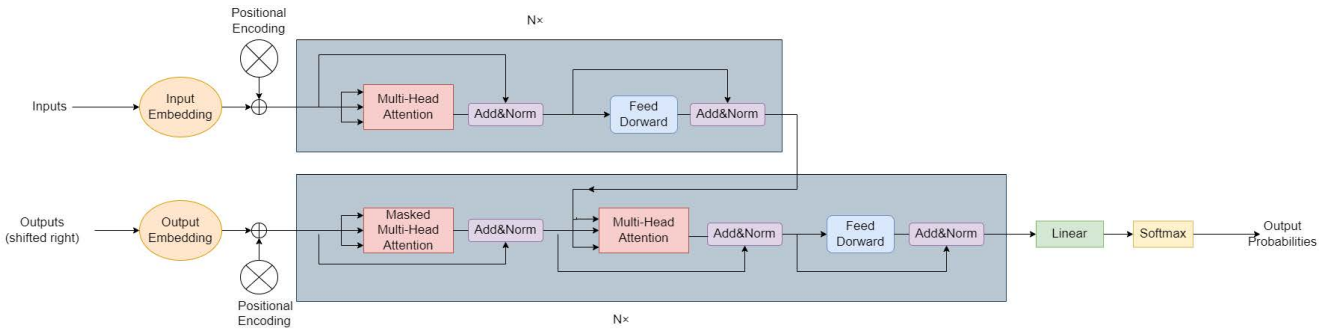

FIGURE 2. Transformer architecture.

As the length of the vector grows longer, the scale range of its dot product result grows wider. As a result, it will fall into the saturation zone after going through the softmax operation, reducing the gradient during backpropagation. It makes the model optimization harder. To address this issue, the inner product value we obtain will be divided by the square root of its length, denoted as d, before executing softmax. Then we get fomula 2.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V \tag{2}$$

In the case that keys, values and queries share the same vector value, the special attention structure is called self-attention.

2) INPUT OF THE TRANSFORMER

In the Transformer, with the word Embedding and the positional Embedding (Positional Encoding) added, the input representation x of a word is constructed. Word Embedding input representation can be produced in a variety of ways, such as pre-trained using Word2Vec, Glove, and so on. This is a common strategy in NLP applications.

We'll also need a new embedding called positional embedding to show the relative position of the word in the phrase. Transformer, unlike RNN, uses all global information but prioritizes the sequential information of words, which is critical for NLP. As a result, we employ Location Embedding to keep track of each word's relative or absolute position in the sequence. With the word Embedding and the position Embedding of the word added, the representation vector x of the word is constructed, and x is used as the input for the Transformer.

3) STRUCTURAL INNOVATION

The output of the Attention module can be multiplexed to combine the information in numerous ways, thereby defining the existing characteristics from several perspectives, to thoroughly investigate the information needed for the model. This is referred to as the **Multi-Head Attention Structure**.

The Encoder block structure of the Transformer is shown in red, and it is made up of Multi-Head Attention, Add and Norm and Feed Forward. We just covered the Multi-Head Attention computing method. Here **Add** refers to $X + MultiHeadAttention(X)$, it is a residual connection that is frequently used in ResNet to overcome the challenge of training multi-layer networks by allowing the network to focus just on the present section of the difference.

Layer Normalization, which is commonly utilized in RNN structures, is referred to as the Norm. Layer Normalization equalizes the mean-variance of the inputs of each layer of neurons, allowing for faster convergence.

The Feed Forward layer is a two-layer completely connected layer with ReLU as the first layer's activation function and no activation function in the second layer.

ReLU ($ReLU(x) = max(0, x)$) is one of the most widely used activation functions in the field of deep learning, and the nonlinear mapping that it can provide is one of the important reasons why deep neural networks can produce excellent results.

B. VIT

Gradually, several researchers attempted to apply the transformer structure, which excels in NLP tasks, to vision tasks, resulting in Vision Transformer, or ViT. Figure 3 shows the architecture of ViT.

The four primary aspects of the ViT process are as follows:

1) Image chunking, that is, to create image patches
2) Image patch embedding and position coding are both available
3) Encoder with Transformer
4) Classification with simple models using features from Encoder.

We will introduce ViT in terms of these four parts.

1) IMAGE CHUNKING

The very first step is pre-processing. Without any special pre-processing, CNN convolves the image in two height and width dimensions natively. To use the Transformer structure, however, we must chunk it first.

If the input is an image x in HWC, and it is now divided into patches of $P \times P \times C$, the total number of patches is $N = HW/P^2$. The dimension of data can be written in the

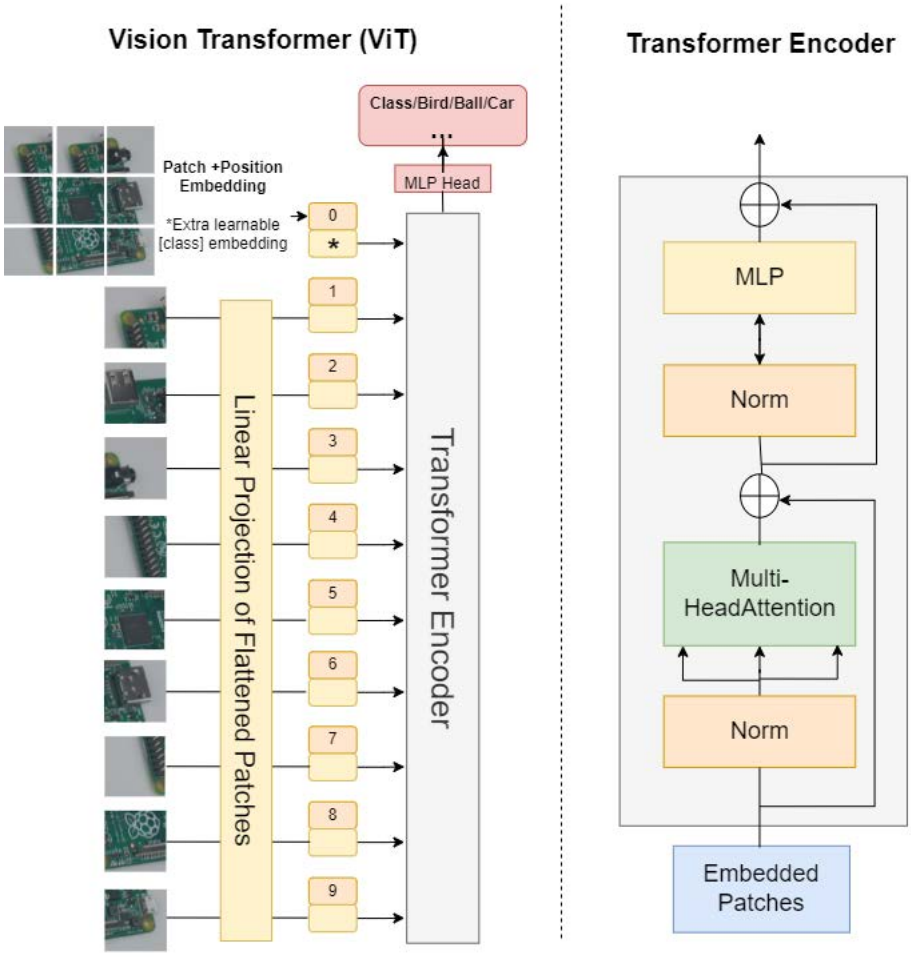

FIGURE 3. ViT architecture.

shape of $N \times P \times P \times C$. After that, each patch is spread, and the data dimension corresponding to it can be expressed as $N \times (P^2 \times C)$. The length of the sequence from the input to the Transformer is N, the number of channels in the input picture is C, and the size of the image patches is P.

2) IMAGE CHUNK EMBEDDING (PATCH EMBEDDING)

Because image chunking is simply a pre-processing step, we must perform an image block embedding operation to convert the vector dimension of $N \times (P^2 \times C)$ into a two-dimensional input of size $N \times D$. Block embedding, like word2vec in NLP, transforms a high-dimensional vector into a low-dimensional representation. The embedding is given a particular code, such as 'CLS' in BERT, to be used as the category prediction result.

After dimensionality reduction, the so-called picture block embedding is essentially a linear transformation, i.e., a fully linked layer, of each spanned patch vector with dimension D.

A position encoding vector must be added to the image block embedding to maintain the spatial location information between the input picture patches. Instead of employing the updated 2D position embedding approach, ViT's position encoding employs a direct 1D learnable position embedding variable, as the authors discovered that in trials, the 2D and 1D embedding methods produced equivalent results.

3) TRANSFORMER

The transformer module is used to extract features, which are then fed into the final classifier to produce the required classification results.

Different investigations have proven that the transformer architecture has a disruptive effect on various tasks by considerably improving the ability to mix features and fuse higher-order knowledge.

4) CLASSIFIER

After enough good features have been retrieved, all that is required for reliable classification results is a simple network topology, and a shallow network like MLP(multilayer perceptron) can achieve incredibly high metrics.

The figures can be transformed into sequence information, which is more conducive to understanding the relationships

between them, thus increasing the metrics, by chunking them and then extracting the features independently.

C. MASK PATCH PREDICTION

The most common front-to-back prediction performed on the input sequence is called unidirectional prediction. In unidirectional prediction, the semantics of the entire utterance is not the focus, which means it is not completely understood. So researchers further proposed bi-directional prediction [27].

In BERT [14], its authors argue that bi-directional prediction still cannot understand the semantics of the whole utterance completely. A better approach is to use the full information of the context to predict **[mask]**. That is, a random part of the content is masked as the target of prediction. BERT uses a two-step operation to perform the training of the mask strategy.

The first step is to mask 15% of the words in an article. According to the context, the model will omnidirectionally predict the masked words. For N articles, average W words in each article and randomly 15% of the words being masked, the task of the model is to predict the $15\% \times W \times N$ masked words correctly. The transformer is trained to get appropriate parameters for the prediction purpose.

Then, the second step is to continue training the parameters of the model. For example, from the above-mentioned N articles, select the utterances. These include two consecutive contextual statements and a discrete pair of statements. The transformer model will then identify consecutive and discontinuous utterances.

In ViT [21], we use a similar strategy called **masked patch prediction** for self-supervision which we believe will make the model training more accurate and accelerate convergence.

For preliminary self-supervision investigations, we use the masked patch prediction objective. To do this, we operate part of patch embeddings in the procedure. Among them all, we replace the embeddings with a learnable [mask] embedding by the amount of 80% while 10% will be a random another patch embedding, and the remaining(10%) are left alone. In the end, we use the respective patch representations to predict the mean color of every color channel for each corrupted patch to recover information.

D. LABEL SMOOTH

With the widespread application of computer vision, there are many adversarial generation methods against the model, which seriously affects the effectiveness of the model and the security of daily use, for example, FGSM [28], BIM [29], JSMA [30] and DeepFool [31]. For such problems, researchers have proposed a series of defenses, and **label smooth** [32] is one of the most widely used.

In classification tasks, we often use one-hot codes to mark the category information of the images. For an N-class classification task, the label Y is an N-d vector. If the corresponding label is K, then only the K-th element of Y is 1 and all others are 0.

When using the label smooth strategy, we distribute a portion of the values corresponding to the true labels, denoted as α, equally to the components of each category, making the differences between categories smoother. In the classification task, the label corresponding to each sample is a vector of length equal to its number of categories, and when the number of categories is 4, a 4-dimensional (4d) vector is used for its representation. Taking $N = 4, \alpha = 0.2$ as an example, if the true category is 2, then the 4d vectors obtained before and after label smooth processing are [0, 1, 0, 0] and [0.05, 0.85, 0.05, 0.05] respectively.

The advantages of label smoothing strategy can be explained from several perspectives:

1) In real scenarios, there could be noise especially when there is a large amount of data. Label smoothing can be added to prevent the model from incorrectly learning the noise.
2) Sometimes we train a model to give fairly high confidence, which may lead to other problems such as over-fit. To address this issue, label smoothing can be introduced to improve the learning difficulty of the model.
3) For image classification, there will be some ambiguous cases. Some pictures (like bottles and bowls) are using soft-target instead of hard-target. Label smoothing can provide a supervised effect for both categories. Here, we denote the original one-hot label as the hard-target, and the result of its label smoothing as the corresponding soft-target.

With label smoothing, our proposed model, LPViT can achieve more robust labeling and enhanced patch correlation.

LPViT model, compared with other classification models, has the following characteristics, which are the reasons why it works better than traditional CNN models and other transformer structures:

- The LPViT model, compared with other CNN-based models, focuses more on the information in the input and extracts the parts that are more relevant to the results, which optimizes the convergence process of the model.
- Mask and label smooth strategies in LPViT are more robust and can achieve better generalization than other transformer-based models for unseen data distributions to obtain better results in different vision tasks, such as detection and classification.

IV. EXPERIMENTS

A. CLASSIFICATION TASK

1) DATASET

We use a dataset called **micro-PCB**, which contains 8,125 photos of 13 micro-PCBs with high resolution. The average width and height of all photos are 1949 and 2126 pixels.

Under optimal lighting circumstances, the micro-PCBs were photographed in 25 various positions relative to the camera. Each micro-PCB was photographed in 5 separate rotations in each position. As a result, each micro-PCB has

TABLE 1. PCBs and labels.

PCB	Label
Raspberry Pi A+	A
Arduino Mega 2560 (Blue)	B
Arduino Mega 2560 (Black)	C
Arduino Mega 2560 (Black and Yellow)	D
Arduino Due	E
Beaglebone Black	F
Arduino Uno (Green)	G
Raspberry Pi 3 B+	H
Raspberry Pi 1 B+	I
Arduino Uno Camera Shield	J
Arduino Uno (Black)	K
Arduino Uno WiFi Shield	L
Arduino Leonardo	M

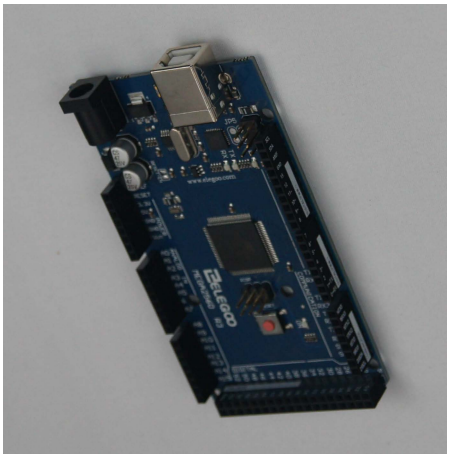

FIGURE 4. Raspberry Pi A+ sample image.

125 different orientations to the camera. For every orientation, a micro-PCB was photographed four times, which were coded for training. After that, a single capture and testing of a micro-PCB of the same make and model was performed. An image of a micro-PCB utilized in training will not be used in testing.

Although the micro-PCBs coded for training and testing are essentially similar, there are relatively minor changes in some circumstances. Each micro-PCB in the dataset comprises 500 training photos and 125 test images, resulting in a 6,500/1,625 train/test split. Figures 4 - 7 demonstrate four samples in the **micro-PCB** dataset.

Each type of PCB and its category (Label) are listed in Table 1.

2) MODEL TRAINING

In our experiments, we utilize Python and Pytorch1.8.0. The GPU GTX2080TI and CPU i7-10875h are the hardware devices we employ to speed up the training.

We chose Adam [33] as our optimizer. The optimum parameters beta1 and beta2 are set to be 0.9 and 0.999, respectively, with a learning rate of 0.0001. To prevent the denominator from being 0, we use eps = 1e-08. The **weight_decay** equals 0, which means we do not take them into the loss. With batch size of 64 we set the binary classification problem and choose **binary_cross_entropy_with_logits** to be the loss function. The total training epochs are 75.

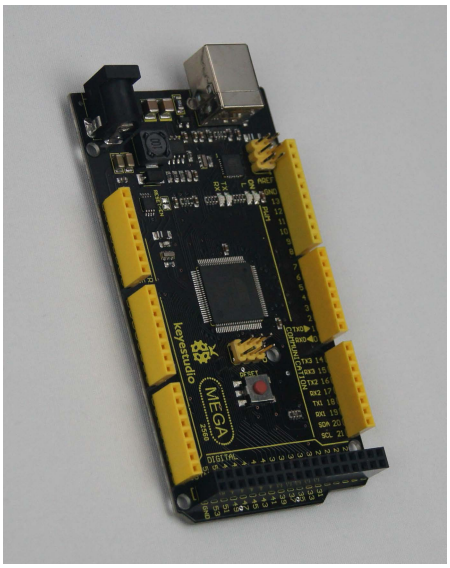

FIGURE 5. Arduino mega 2560 (black) sample image.

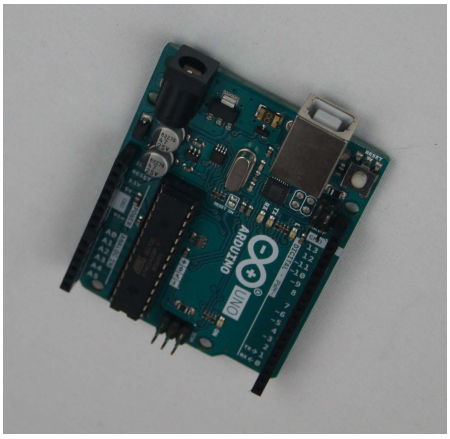

FIGURE 6. Beaglebone black sample image.

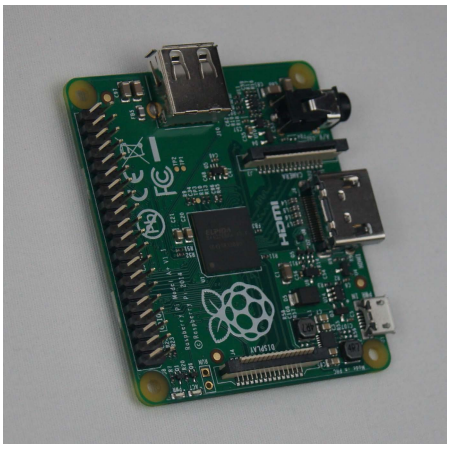

FIGURE 7. Arduino Leonardo sample image.

To further improve the training effect of the model, the learning rate is dynamically adjusted, and we use a learning rate decay strategy to reduce the learning rate by 10% for

TABLE 2. Performance of different model Configs for classification on the micro-PCB dataset.

Model	Attention	MPP	Label Smooth	Accuracy	Precision	Recall	F1-score
ResNet50(Baseline)	×	×	×	0.977236	0.978116	0.980741	0.979426
ViT	✓	×	×	0.984562	0.983261	0.981198	0.984294
ViT + MPP	✓	✓	×	0.987702	0.987086	0.986462	0.986467
LPViT(ViT + Label Smooth + MPP)	✓	✓	✓	**0.990769**	**0.990288**	**0.990194**	**0.990335**
Swin Transformer	✓	×	×	0.990106	0.983261	0.981198	0.984294

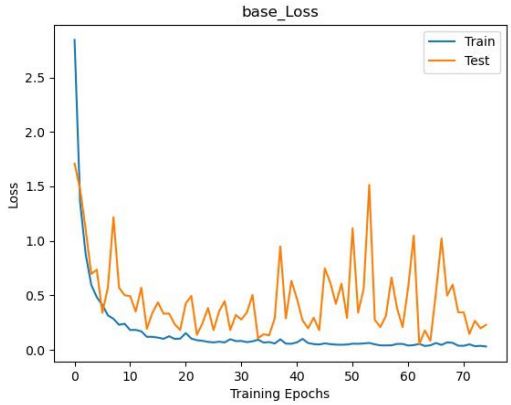

FIGURE 8. Base model loss.

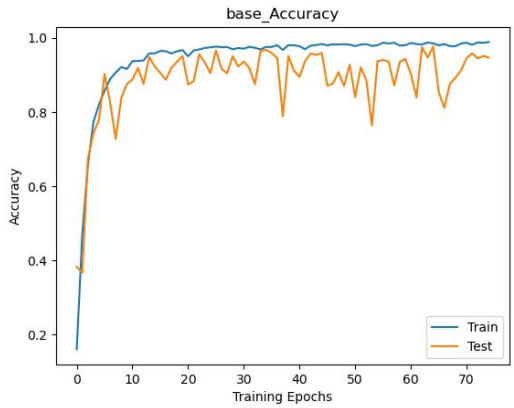

FIGURE 9. Base model accuracy.

every 30 rounds of training. The initial learning rate is set to 0.0001.

The probability of using tokens in masked prediction tasks **mask_prob** is set to be 0.15, and the probability of randomly replacing is set to be **random_patch_prob** = 0.30, with **replace_prob** = 0.50, which means half of the tokens are set to be mask token.

3) DATA AUGMENTATION

We utilize a random crop, a random horizontal flip with a chance of 0.5, and scale the input photos to a fixed size of 224 × 224 for the training set. For test data, there is no data augmentation.

4) PERFORMANCE METRICS

In this research, we use accuracy, precision, recall, and F1-score as the metrics to evaluate the performance of different models. They are calculated using formulas 3, 4, 5, and 6.

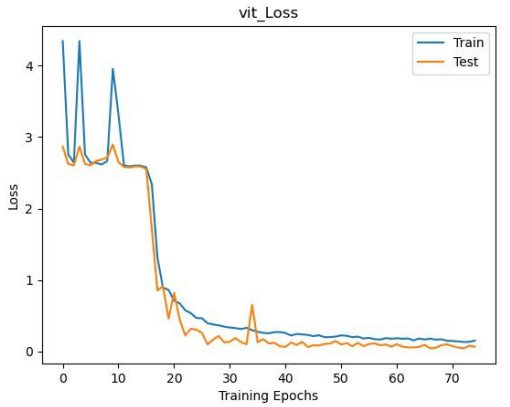

FIGURE 10. ViT loss.

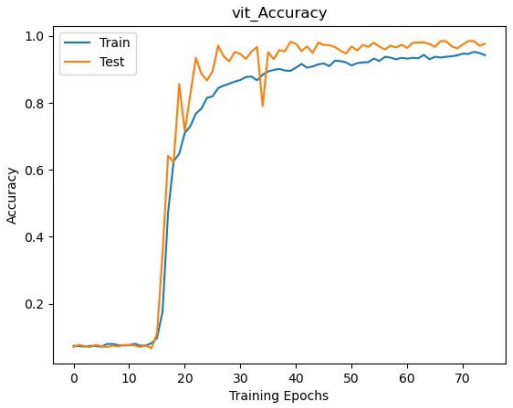

FIGURE 11. ViT accuracy.

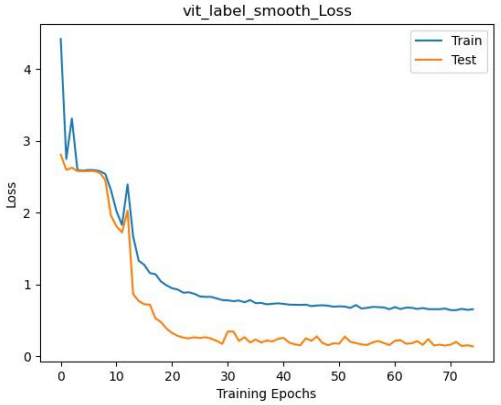

FIGURE 12. ViT label smooth loss.

We denote the total number of samples as N, the number of correctly classified samples as N^c, the predicted number of positive samples as $\hat{N}_+$, where the true number of positive samples is N_+^c and the number of positive samples in the data

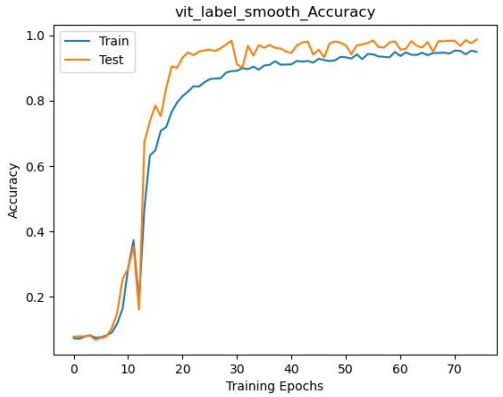

FIGURE 13. ViT label smooth accuracy.

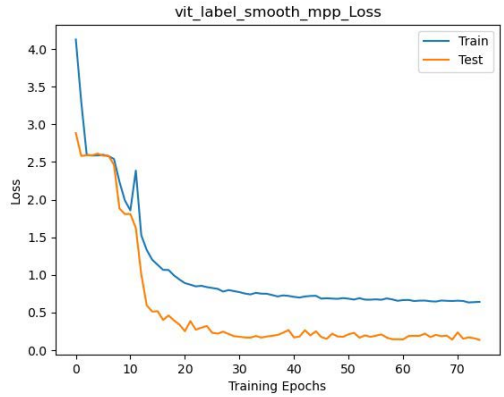

FIGURE 14. LPViT loss.

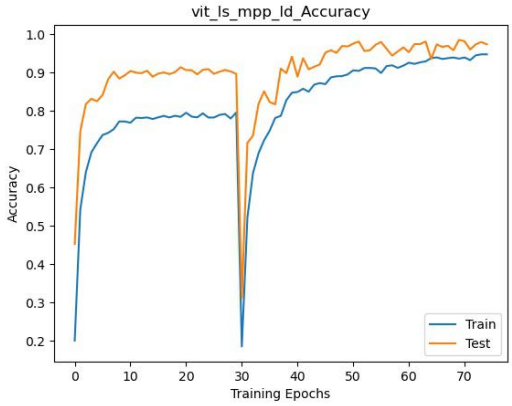

FIGURE 15. LPViT accuracy.

set is N_+.

$$Accuracy = \frac{N^c}{N} \tag{3}$$

$$Precision = \frac{N_+^c}{\hat{N}_+} \tag{4}$$

$$Recall = \frac{N_+^c}{N_+} \tag{5}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

5) EXPERIMENTAL RESULTS

We choose ResNet50 [34] as the baseline model for this experiment, which has shown excellent performance in many vision tasks. A comparison with it can effectively illustrate the performance of the proposed model.

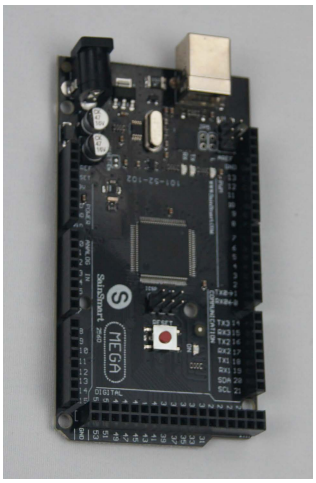

FIGURE 16. Error sample 1.

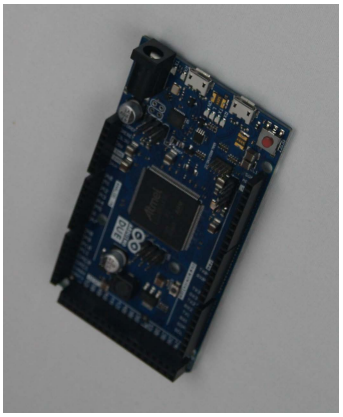

FIGURE 17. Error sample 2.

TABLE 3. Defect types and numbers.

Defect Type	Trainval Set	Test Set
open	1149	553
short	924	393
mousebite	1258	490
spur	1047	398
copper	927	394
pin-hole	927	393

We use three different configurations for training.

- ViT.
- ViT + Label Smooth.
- LPViT

The metrics of the baseline model and three ViT-based methods are shown in Table 2. It can be found that our proposed model improves the accuracy rate by 1.36% compared to the baseline, which fully illustrates the effectiveness of the proposed model and its adaptability to the dataset.

In the first model configuration with the ViT only, the performance has already exceeded the baseline model. After adding label smooth and MPP (mask patch prediction), the accuracy can be improved by another 0.62%. They reduce

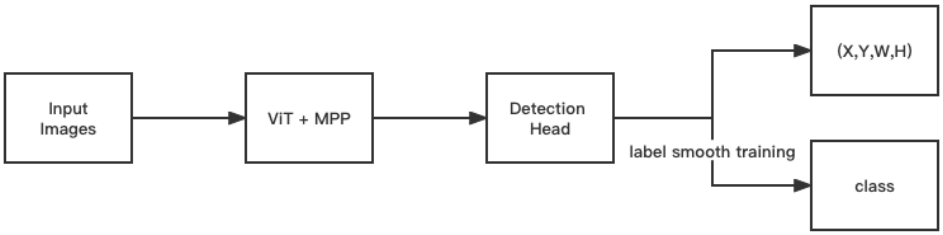

FIGURE 18. Overview of LPViT for defect detection.

TABLE 4. Metrics of different model Configs for DeepPCB dataset detection.

Model	Attention	MPP	Label Smooth	AP	AP^{50}	AP^{75}
Faster RCNN ResNet101(Baseline)	×	×	×	0.909611	0.915368	0.907254
Faster RCNN ViT	✓	×	×	0.846540	0.881967	0.871245
Faster RCNN ViT + MPP	✓	✓	×	0.975138	0.983691	0.970813
Faster RCNN LPViT(ViT + Label Smooth + MPP)	✓	✓	✓	**0.976322**	**0.988376**	**0.976661**

the risk of overfitting and enhance the smoothness of the boundary, which improves the model's understanding of the relationship between different regions of the images. Precision, Recall, and F1-score demonstrate a similar pattern of improvement. We also compare our model with another transformer Swin Transformer model on the classification task. As shown in Table 2, LPViT, outperforms other models by all metrics.

Furthermore, the processing speed of our model is 156.1FPS (frame per second), while the processing speed of the Swin Transformer is 129.9FPS.

Figures 8 - 15 demonstrate the accuracy and loss versus epochs for each model. It should be specifically noted that in all ViT-based methods, the accuracy on the training set outperforms the test set after a certain number of epochs of training. It indicates that the patch method of ViT can better extract the association of different regions in the image and obtain the underlying information in the recognition process to achieve better migration results on unseen new data.

6) ERROR ANALYSIS

Figures 16 and 17 provide two examples of misclassification cases. With a further examination of the misclassified samples, it can be seen that the images are blurred to a certain extent compared to the correctly classified images. Some relevant details for classification are lost, resulting in the wrong classification.

In the future, we can model the correlation between different parts of the images and use it to recover the missing data of a certain place.

B. DEFECT DETECTION

In order to fully investigate the effectiveness of our proposed model for different tasks, we use it to perform defect detection tasks. It should be noted that in order to maintain the relative position of the original images, we turn the output patch sequence back to the original two-dimensional arrangement as the output of the backbone. Figure 18 illustrates the overview of the system for defect detection.

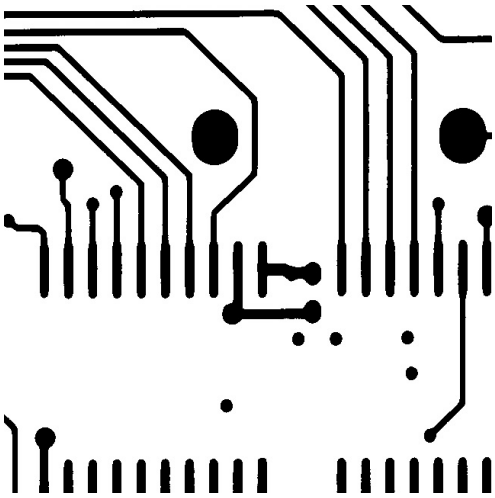

FIGURE 19. PCB template sample.

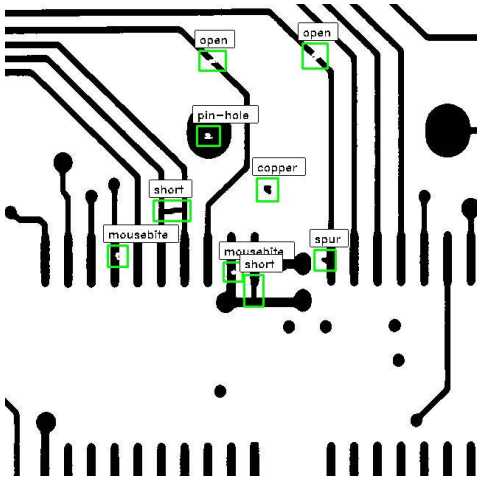

FIGURE 20. PCB defect annotation sample.

1) DATASET

Here we use a new dataset **DeepPCB** [3] for detection experiments. It contains 1500 pairs of images. In each pair, there is one defect-free template image and an annotated defect image. In **DeepPCB**, there are six different types of PCB

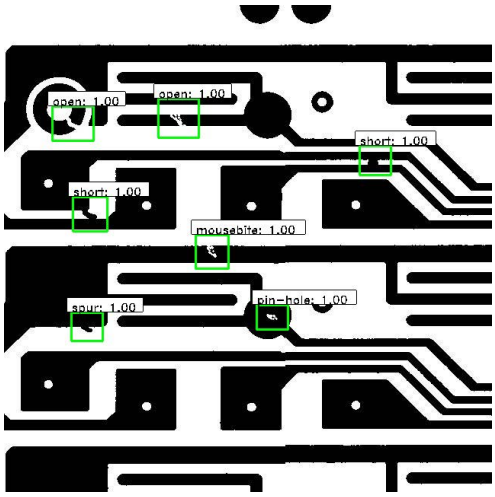

FIGURE 21. Detection result sample 1.

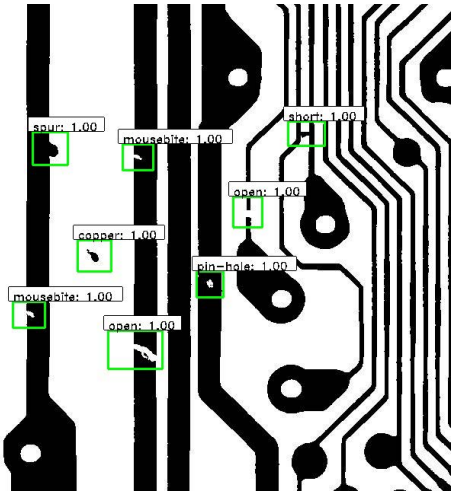

FIGURE 22. Detection result sample 2.

defects: open, short, mousebite, spur, pin hole, and spurious copper. The details of each type are shown in Table 3. Figures 19 and 20 demonstrate an example pair of images.

2) MODEL TRAINING

We continue to use Adam as the optimizer, with a learning rate of 0.0005 for the defect detection experiment. We set the total training epochs to be 2000.

3) PERFORMANCE METRICS

We use **mean Average Precision(mAP)** as the metric for PCB defect detection task.

We arrange the detection results predicted by the trained model in decreasing order of the confidence score. If the predicted box and the ground truth with the same pair get an **IoU** (intersection over union) greater than a certain threshold, we believe it is a ''match''. We get mAP by fusing different AP values. Here we use AP, AP^{50}, and AP^{75} which have different sampling and threshold values. The calculation process is the same and they can evaluate the effectiveness of the model from different perspectives.

4) EXPERIMENTAL RESULT

We chose Faster RCNN [35] as our baseline model, which is a very widely used two-stage detector with outstanding detection results. We modify its backbone step by step to be LPViT to test the effect of the model on the defect detection task.

After replacing the backbone with ViT, there is a significant drop for each AP. It could be caused by the loss of some important interaction information during one-dimensional sequence processing, which makes the extracted features not the most relevant part of the detection task. The drop is significantly improved by adding MPP, which demonstrates the effect of the MPP strategy on model learning. MPP obviously improves the learning direction of the model and has significant effects on both classification and detection tasks. The addition of label smoothing further reduces some of the false positives and improves the performance of the model. Table 4 shows the comparative experimental results with the maximum values in each column identified. LPViT improves all metrics by more than 6%. It also outperforms the current SOTA model (98.8% vs. 98.6%) that employs the group pyramid pooling module [3]. Furthermore, the processing speed of our model is 27.7FPS, while the processing speed in [3] is 24.6FPS. The experimental results fully demonstrate the effectiveness of the proposed structure.

V. CONCLUSION

PCB devices have greatly facilitated human life. The classification and defect detection of PCB image data can greatly accelerate downstream tasks such as production and sorting, effectively improve manufacturing and recycling efficiency, and meet the growing demand for PCBs. Both industry and research institutions have devoted great efforts to the classification and defect detection of PCBs, which requires full consideration of the small scale of the PCB model, the small distinction between categories, and various imaging conditions such as image illumination angles. Among the massive PCBs, a big group of applications is using micro-PCBs manufactured by several prevailing companies. Once the makes and models of PCBs are identified, the sub-components could be easily retrieved through a bill of materials. In this research, we propose a transformer-based classification model, LPViT, which can be used to classify micro-PCBs based on their makes and models and detect defects on PCBs. We also introduce mask patch prediction and label smooth to improve the accuracy and the robustness of the model. Through comparative experiments, our proposed model has achieved SOTA performance on the micro-PCB dataset for classificaiton and on DeepPCB dataset for defect detection.

Meanwhile, there are some limitations in this research, which could be further studied in future research. Data augmentation can be achieved by different methods. In the future, we could apply more complex data augmentation models to further enrich the dataset, which could potentially promote the performance of our system. The dataset we use for classification is micro-PCBs. The classification performance of

our system on large-scale PCBs needs to be further studied and verified.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[2] A. Byerly, T. Kalganova, and A. J. Grichnik, "On the importance of capturing a sufficient diversity of perspective for the classification of micro-PCBs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 209–219.

[3] S. Tang, F. He, X. Huang, and J. Yang, "Online PCB defect detector on a new PCB defect dataset," 2019, *arXiv:1902.06197*.

[4] P. S. Malge and R. S. Nadaf, "PCB defect detection, classification and localization using mathematical morphology and image processing tools," *Int. J. Comput. Appl.*, vol. 87, no. 9, pp. 40–45, Feb. 2014, doi: 10.5120/15240-3782.

[5] K. Kamalpreet. (2014). *PCB Defect Detection and Classification Using Image Processing*. [Online]. Available: https://www.semanticscholar.org/paper/PCB-Defect-Detection-and-Classification-Using-Image-Kamalpreet/592fd41d58ac50a38da9a8d9dd55ad34ec3e1a00

[6] J. Kim, J. Ko, H. Choi, and H. Kim, "Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder," *Sensors*, vol. 21, no. 15, p. 4968, Jul. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/15/4968

[7] U. Ankit. (Aug. 2021). *Image Classification of PCBs and Its Web Application (Flask)*. [Online]. Available: https://towardsdatascience.com/image-classification-of-pcbs-and-its-web-application-flask-c2b26039924a

[8] C. Yang. (2020). *Machine Learning and Computer Vision for PCB Verification*. [Online]. Available: https://kth.diva-portal.org/smash/get/diva2:1529213/FULLTEXT01.pdf

[9] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, and N. Zheng, "End-to-end object detection with fully convolutional network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 15849–15858.

[10] T. Zhou, D. Fan, M. Cheng, J. Shen, and L. Shao, "RGB-D salient object detection: A survey," *Comput. Vis. Media*, vol. 2021, pp. 1–33, Jan. 2021.

[11] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 30, 2021, doi: 10.1109/TITS.2021.3096854.

[12] Z. Zhou, L. Li, A. Fürsterling, H. J. Durocher, J. Mouridsen, and X. Zhang, "Learning-based object detection and localization for a mobile robot manipulator in SME production," *Robot. Comput.-Integr. Manuf.*, vol. 73, Feb. 2022, Art. no. 102229.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–12.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[15] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[17] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1441–1451.

[18] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "ERNIE 2.0: A continual pre-training framework for language understanding," 2019, *arXiv:1907.12412*.

[19] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 213–229.

[20] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," 2021, *arXiv:2012.15840*.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[22] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2020, *arXiv:2012.12877*.

[23] T. Zhou, S. Ruan, and S. Canu, "A review: Deep learning for medical image segmentation using multi-modality fusion," *Array*, vols. 3–4, Sep. 2019, Art. no. 100004, doi: 10.1016/j.array.2019.100004.

[24] Y. Shachor, H. Greenspan, and J. Goldberger, "A mixture of views network with applications to multi-view medical imaging," *Neurocomputing*, vol. 374, pp. 1–9, Jan. 2020, doi: 10.1016/j.neucom.2019.09.027.

[25] V. Q. Joe and P. L. Westesson, "Tumors of the parotid gland: MR imaging characteristics of various histologic types," *Amer. J. Roentgenol.*, vol. 163, no. 2, pp. 433–438, Aug. 1994.

[26] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

[27] Z. Huang, X. Wei, and Y. Kai, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.

[28] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[29] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*.

[30] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 372–387.

[31] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[32] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 558–567.

[33] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

KANG AN received the master's degree in circuit and system from the Guilin University of Electronic Technology, in 2007. He joined Hangzhou Normal University as a Lecturer. His research interests include the Internet of Things technology and machine learning.

YANPING ZHANG received the M.S. and Ph.D. degrees in computer science from The University of Alabama, in 2009 and 2012, respectively. She joined the Department of Computer Science, Gonzaga University, as an Assistant Professor, in 2012. She is currently an Associate Professor of computer science at Gonzaga University. Her research interests include but are not limited to the Internet of Things (IoT), smart and intelligent systems, machine learning, wireless communication, and computer security.