Applied Computing and Intelligence

https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
4(2): 253–268.
DOI: 10.3934/aci.2024015
Received: 30 October 2024
Revised: 19 December 2024
Accepted: 19 December 2024
Published: 26 December 2024

Research article

MAE-GAN: a self-supervised learning-based classification model for cigarette appearance defects

Youliang Zhang, Guowu Yuan*, Hao Wu and Hao Zhou

School of Information Science & Engineering, Yunnan University, Kunming 650504, China

* **Correspondence:** Email: gwyuan@ynu.edu.cn.

Academic Editor: Zhiling Long

Abstract: Appearance defects frequently occur during cigarette manufacturing due to production equipment or raw materials. Appearance defects significantly impact the quality of tobacco products. Since manual inspection cannot keep pace with the demands of high-speed production lines, rapid and accurate automated classification and detection are essential. Supervised learning is predominantly employed in research on automated classification of product quality appearance defects. However, supervised learning necessitates substantial labeled data for training, which is time-consuming to annotate and prone to errors. This paper proposes a self-supervised learning-based classification model for cigarette appearance defects. This is a generative adversarial network (GAN) model based on masked autoencoders (MAE), called MAE-GAN. First, this model combines MAE as a generator with a simple discriminator to form a generative adversarial network according to the principle of mask reconstruction in MAE. The generator reconstructs the images to learn their features. Second, the model also integrates MAE's loss function into the GAN's loss function. This lets the model focus on pixel-level losses during training. As a result, model performance is improved. Third, a Wasserstein GAN with gradient penalty (WGAN-GP) is added to stabilize the training process. In addition, this paper preprocesses cigarette images through segmentation and recombination. Neural networks typically accept images with the same width and height. Due to the narrow shape of cigarette images, if the image is directly transformed into a square and fed into a neural network, the details of the image will be severely lost. This paper segments the cigarette image into three parts and recombines them into images with similar length and width, greatly improving classification accuracy.

Keywords: cigarette; appearance defect; classification; MAE; generative adversarial network; self-

supervision

1. Introduction

China is the leading global tobacco production and sales country, with Yunnan Province being the foremost tobacco-producing region. Manufacturers emphasize the quality of tobacco products, with particular attention to the appearance of cigarettes as a critical aspect of quality control. During the production process, cigarettes are prone to developing stains, wrinkles, and other defects. Historically, manual sampling was the primary method for detecting these appearance defects. However, manual detection has become impractical given that production lines now operate at speeds of 150–200 cigarettes per second.

With advancements in computer vision, automatic defect detection has been increasingly utilized in quality control processes. In cigarette appearance defect research, Yuan et al. [1] employed transfer learning and multi-scale learning within the ResNeSt model, modifying the model's activation function to enhance defect detection capabilities. Qu et al. [2] proposed an improved single-shot MultiBox detector (SSD) model to address the challenge of low detection efficiency. Additionally, Liu et al. [3] incorporated channel attention mechanisms and refined the loss function in the YOLOv5s model to improve defect detection performance further. Yuan et al. [4] applied an improved YOLOv4 model for defect detection. They added SE attention mechanisms, modified activation functions, and adjusted the K-means method to speed up model convergence. Liu et al. [5] used an improved CenterNet model. They introduced CBAM, deformable convolutions, and a feature pyramid, improving activation functions and data augmentation. Ma et al. [6] proposed the CJS-YOLOv5n model. They added the C2F module and Jump Concat to the YOLOv5n model. They also modified the loss function to achieve faster detection speeds.

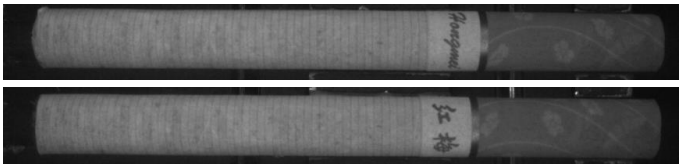
However, the existing detection methods have two disadvantages: (1) Cigarettes are elongated, while neural networks typically require square images. Compressing cigarette images horizontally results in severe detail loss. These methods do not account for the unique dimensions of cigarette images. (2) These methods rely on supervised learning, which requires labeled data. Labeling data is costly and can be inaccurate. Self-supervised learning can train models from unlabeled data, reducing dependence on labeled data [7,8]. Additionally, self-supervised learning can learn valuable features across multiple tasks. Transferring the learned features to different domains or tasks is more accessible, usually with good results [9,10].

This study proposes a self-supervised learning-based classification model for cigarette appearance defects called MAE-GAN. First, the model uses MAE as a generator to form a GAN. The generator, MAE, reconstructs images to learn their features. Second, MAE's loss function is integrated into the GAN's loss function. This helps the model focus on pixel-level losses during training, improving the model's effectiveness. Finally, this model adds Wasserstein distance and gradient penalty from WGAN-GP to stabilize the training process. The experiments show that the proposed model can significantly improve classification performance. This model outperforms MAE in downstream tasks. This model improves by 3.3% in fine-tuning tasks compared to the original model. In linear probing, it also improves by 4.4%. In addition, this paper attempts to segment and recombine cigarette images into images with similar length and width in preprocessing, solving the problem of slimness in cigarette images.

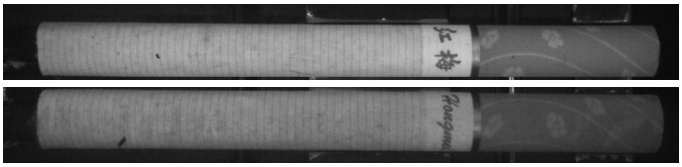
2. Introduction and preprocessing of cigarette appearance defect dataset

2.1. Introduction to cigarette appearance defect dataset

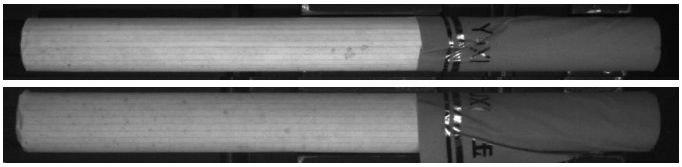
Yunnan China Tobacco Industrial Co., Ltd provided the dataset for cigarette appearance defects. High-speed industrial cameras captured cigarette images on an automated production line. The cameras capture the front and back of each cigarette at different positions on the production line. The two images capture most details of the appearance of the cigarette. Based on the actual needs of the cigarette factory, the appearance defects were categorized into five types: normal, spots, wrinkles, misaligned teeth, and no filter [11], as shown in Figure 1. Spots refer to black dots or stains of varying degrees on the cigarette's appearance. Wrinkles refer to wrinkle-like shapes on the surface of the cigarette. Misaligned teeth occur when the cigarette's wrapping paper is not aligned with the tobacco flow. No filter occurs when the filter is not correctly adhered to the tipping paper, leading to detachment.



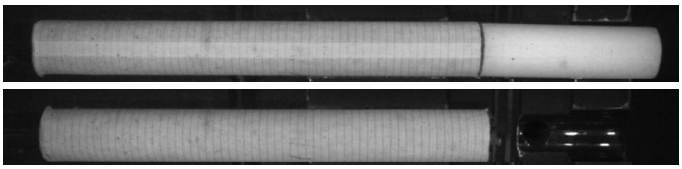
(a) Normal



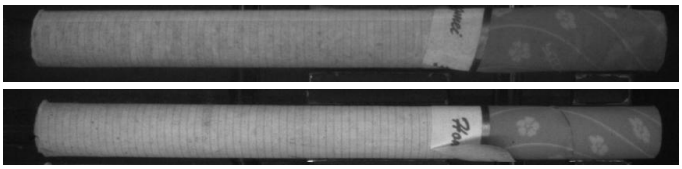
(b) Dotted



(c) Folds



(d) No filter



(e) Misaligned teeth

Figure 1. Examples of cigarette appearance defects.

Since some categories had fewer images, data augmentation methods were applied. These methods included flipping, cropping, and affine transformations. After augmentation, the cigarette dataset consisted of a total of 6400 images.

2.2. Cigarette image segmentation and recombination

The standard length of a cigarette is 84 mm, and its diameter is 7.8 mm. The collected cigarette images have an aspect ratio of approximately 10:1. The cigarette images are elongated in shape. Neural networks typically require square input images. If the elongated cigarette images are reshaped into squares, a lot of detail is lost, which can affect classification accuracy.

Previous studies did not consider this unique feature of cigarette images. Compressing the elongated cigarette images into equal dimensions causes deformation. This makes some cigarette defects challenging to detect. Therefore, this study attempts to segment and recombine the cigarette images. This method ensures that the content in the image does not undergo severe deformation when input to the network.

Suppose the cigarette sample image is $im(x,y)$, where H is the height of the cigarette sample image, W is the width, and the reconstructed image is $im_{recomb}(x,y)$. The method is described in Algorithm 1.

Algorithm 1. Cigarette image segmentation and recombination algorithm.

Input: Original cigarette image $\{im(x,y)/0 \le x < H, 0 \le y < W\}$.

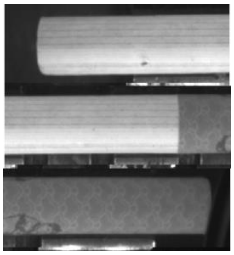
Output: Segmented and recombined cigarette image $\{im_{recomb}(x,y)/0 \le x < 3*H, 0 \ge y < W/3\}$.

Step1： The cigarette image is evenly segmented into three parts, as shown below: $\{im_{left}(x,y)/0 \le x < H, 0 \le y < W/3\}$, $\{im_{mid}(x,y)/0 \le x < H, W/3 \le y < 2*W/3\}$, and $\{im_{right}(x,y)/0 \le x < H, 2*Width/3 \le y < W\}$, as show in Figure 2b.

Step2： Splice im_{left}, im_{mid}, im_{right} at the long side in the order of top, middle and bottom to form a nearly square image $\{im_{recomb}(x,y)/0 \le x < 3*H, 0 \ge y < W/3\}$ as show in Figure 2c.

(a) Original cigarette image.

(b) Cigarette image after horizontal segmentation.



(c) Cigarette image after recombination.

Figure 2. Example of segmentation and recombination of cigarette sample images.

After segmentation and recombination, the cigarette image's aspect ratio changes from approximately 1:10 to about 1:1.1. With the total pixel count unchanged, more horizontal detail is preserved, ensuring higher accuracy in subsequent object classification.

3. Methods

3.1. Motivation of the method

Self-supervised learning (SSL) has been studied in vision for a long time [12]. It pertains to models without manual annotation labels, enabling the utilization of vast unlabeled data. SSL offers competitive results compared to supervised pre-training baselines in various downstream tasks, including image classification [13], object detection [14], and semantic segmentation [15].

MAE [16], a prominent branch in self-supervised learning, has garnered widespread attention. It learns image features by masking random patches of input images and reconstructing the missing pixels. This method is similar to the generative adversarial networks (GANs) generator, as both derive images from input information, albeit with distinct inputs. MAE infers missing pixels based on existing ones, whereas GAN generators produce samples resembling real data from random noise or other inputs through a learning process [17]. Capitalizing on this similarity, this paper utilizes MAE as a generator to construct a generative adversarial network.

MAE uses dense loss as its loss function. This function performs well and considers global loss. However, it cannot work alone as a GAN loss function. GAN loss functions focus on critical features [18], lacking consideration of global loss. This paper combines both loss functions to improve training. In a GAN, the discriminator detects the authenticity of fake images by extracting key features [19–21]. The generator focuses on creating these key features to fool the discriminator. However, this limits the generator's ability to learn other areas. As a result, some information may be missed, leading to poor training. MAE's loss function works differently. It uses dense loss over image blocks to learn each image's representation. MAE's loss function considers global information and computes global pixel loss. It also aligns semantics and spatial sensitivity effectively. Therefore, this paper combines dense loss with the GAN loss. This improves the model's performance. Based on this, we combine the discriminator's loss on the authenticity of fake images with MAE's dense loss. Together, they form a new GAN loss function.

In summary, this paper proposes the MAE-GAN model, a generative adversarial network consisting of a generator (MAE) and a discriminator. The MAE-GAN model employs a dual-network game between the generator and the discriminator to generate realistic data. Meanwhile, this paper improves the adversarial network's loss function by introducing a dense loss function on its basis, requiring the generator to consider not only fooling the discriminator but also minimizing the dense loss between the generated and real images. This paper incorporates the Wasserstein distance and gradient penalty from WGAN-GP during the training process to ensure smoother and more stable training.

3.2. Proposed model

The MAE-GAN model in this paper consists of a generator (MAE) and a discriminator. The generator (MAE) creates realistic data, while the discriminator distinguishes between real and

generated data. This adversarial training process helps the generator improve data realism. Meanwhile, the discriminator evolves to identify real versus fake data better. Figure 3 shows the proposed model's structure.

Section 3.2.1 introduces the generator module used for classifying cigarette appearance defects. Section 3.2.2 introduces the discriminator module. Section 3.2.3 introduces the improved loss function.

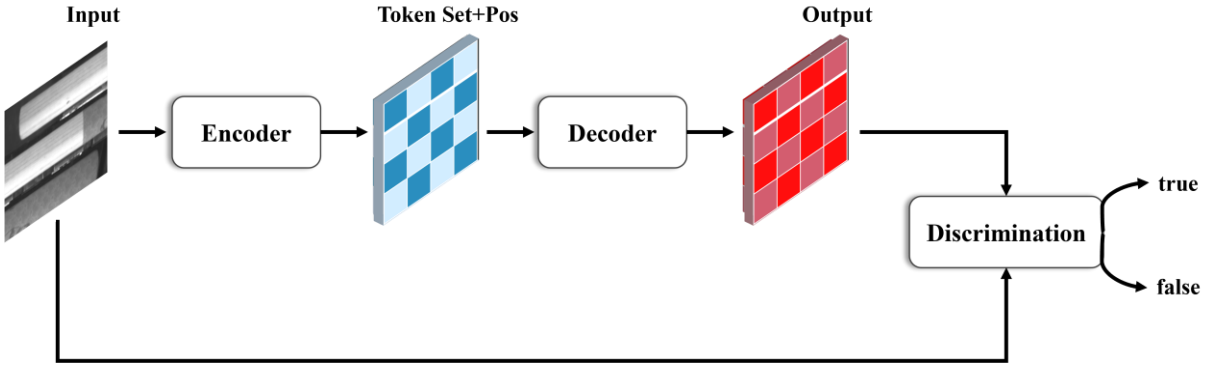


Figure 3. MAE-GAN model for classifying cigarette appearance defects.

3.2.1. Generator module

In this paper, the generator comprises a base version of MAE (as shown in Figure 4). MAE primarily consists of an encoder and a decoder. During the encoder stage, for any batch of images $x_i \in R^{N\times H\times W\times 3}$, after data augmentation, the generator will divide the augmented images into several regular, non-overlapping patches $x_i^{m,n} = x_i(\frac{mW}{p}:\frac{(m+1)W}{p},\frac{nW}{p}:\frac{(n+1)W}{p})$. Some of these patches $x_i^{m,n}$ will be randomly masked out according to a masking rate, while the remaining patches are fed into a linear projection layer. Subsequently, position embedding is added based on their relative positions, serving as the input for the encoder to extract features, forming feature representations with each patch denoted as $\dot{x}_i^{m,n}(\dot{x}_i^{m,n} \in R^{N\times head\times patch\times hiden})$.

$$\dot{x}_i = f_{encoder}(x_i), \dot{x}_i^{m,n} \in \dot{x}_i, \tag{1}$$

where *head* refers to the number of heads in MAE, *patch* represents the feature dimension of each patch, and *hidden* denotes the dimension of the hidden layer output by the MAE encoder.

In the decoder stage, the feature representations $\dot{x}_i$ and mask tokens y_i generated by the encoder form a set $\{\dot{x}_i, y_i\}$. Each element in this set is appended with position encodings pos to supplement positional information. These are then fed into the decoder to reconstruct the masked pixels. Finally, the decoder's output passes through a linear projection layer to transform the information into dimensions compatible with the loss function.

$$\ddot{\mathrm{x}} = f_{decoder}(\{\dot{x}_i + pos, y_i + pos\}), \ddot{\mathrm{x}} \in R^{N\times L\times(patch*2*3)}. \tag{2}$$

In this paper, the predictions generated by MAE are used as input to the discriminator. However, the dimension $(N, L, patch*2*3)$ of the predictions $\ddot{x}_i$ after passing through the final linear projection layer does not match the dimension $(N, 3, H, W)$ of the image x_i. Therefore, a dimension

transformation is required to convert the predictions to $(N,3,H,W)$. This process is called unpatching, which primarily involves reverse calculations based on the original image size, patch size, and stride during segmentation. Specifically, the position of each patch in the original image is determined based on its position in the output image and the segmentation stride, and then it is placed back accordingly. Finally, all the patches are reassembled to obtain an image with the same shape as the original image.

$$\hat{x} = f_{unpatch}(\ddot{x}), \hat{x} \in R^{N\times 3\times H\times W} . \tag{3}$$

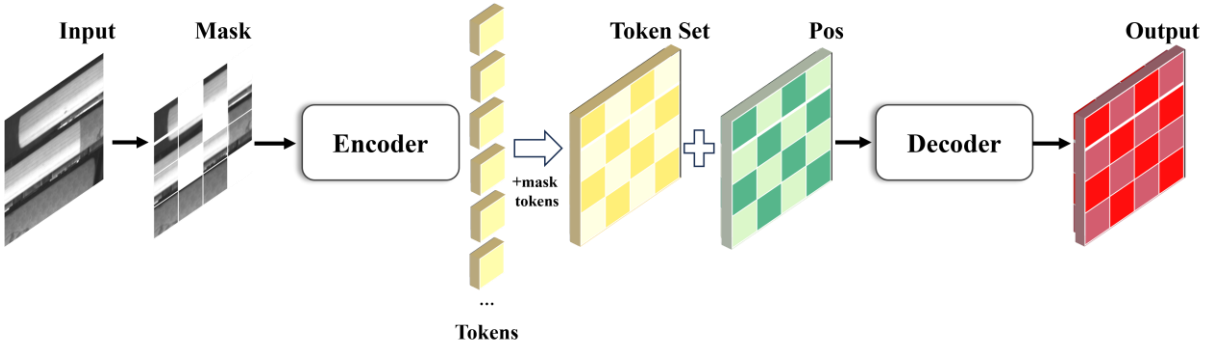


Figure 4. Generator module for classifying cigarette appearance defects.

3.2.2. Discriminator module

This paper uses a feedforward neural network structure as the discriminator. To meet the input requirements, each batch of size is flattened. The images are flattened into a one-dimensional vector $(N,3*H*W)$. The neural network consists of multiple linear layers. Each linear layer is followed by a leaky ReLU $(N,3,H,W)$ activation function. This helps mitigate the vanishing gradient problem. The structure of the discriminator is shown in Figure 5.

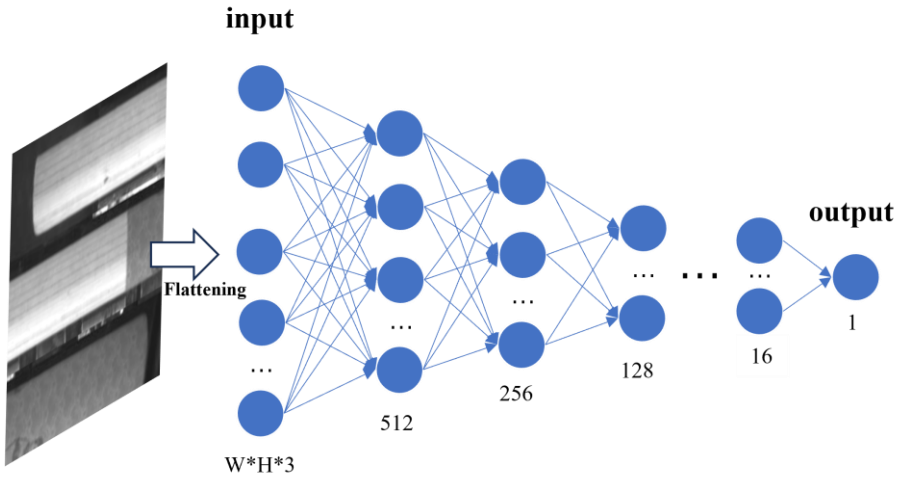


Figure 5. Discriminator module for classifying cigarette appearance defects.

The architecture of the discriminator is as follows. The first linear layer has an input size of

$(N, 3*H*W)$ and an output size of 512. The second linear layer has 512 input nodes and 256 output nodes. Each subsequent layer halves the number of output nodes. This continues until the second-to-last layer, which has 32 input nodes and 16 output nodes. Since the discriminator only needs to classify images as real or fake, it is treated as a binary classification problem. Therefore, the final fully connected layer has 16 input nodes and 1 output node.

This hierarchical structure choice helps gradually reduce network complexity, facilitating better learning of high-level representations of input data. Additionally, using the leaky ReLU activation function helps mitigate the vanishing gradient problem, making the network more robust.

3.2.3. Improved loss function

The loss function used by MAE mainly includes reconstruction error. Reconstruction error measures the difference between the reconstructed masked parts and the real image. With this loss function, MAE effectively trains semantic alignment and spatial sensitivity. This paper integrates MAE's loss function into the GAN loss function. The goal is to improve the model's performance. MAE's loss function trains semantic alignment and spatial sensitivity well. Therefore, this paper incorporated it into the GAN loss function. x_i represents the real image, while $\hat{x}_i$ is the prediction generated by the generator. The MAE loss function formula is as follows:

$$L_M = \frac{1}{M}\sum_{i=1}^{M}\left|x_i - \hat{x}_i\right|. \tag{4}$$

This paper adds the MAE's dense loss function to the original generator's loss function. The formula is as follows:

$$\mathrm{L}_G = -D\left(G\left(x\right)\right) + \gamma L_M, \tag{5}$$

where γ is a hyperparameter used to adjust the impact of the dense loss on the generator's loss function, with a default value of 1.

This paper adds the Wasserstein distance and gradient penalty from WGAN-GP to the discriminator's loss function. Traditional GANs use metrics like JS divergence or KL divergence. However, these metrics can sometimes be unstable. Wasserstein distance addresses this issue. It measures the transport cost between two distributions or the minimal cost to convert one distribution to another. This stabilizes the training process. The gradient penalty limits the magnitude of gradients in the discriminator. This prevents excessive gradient growth, helping stabilize training. It also mitigates gradient explosions or vanishing gradients between the generator and discriminator. Similarly, we introduce MAE's dense loss into the discriminator's loss function. The formula for the discriminator's loss is as follows:

$$L_D = -E_{x\sim P_{data}}\left[D\left(G\left(x\right)\right)\right] + E_{x\sim P_{data}}\left[D\left(G\left(x\right)\right)\right] + \alpha E_{\hat{x}\sim P_{\hat{x}}}\left[\left(\left\|\nabla_{\hat{x}} D\left(\hat{x}\right)\right\|_2 - 1\right)^2\right] + \gamma L_M, \tag{6}$$

where α controls the effect of gradient penalty, defaulting to 10. γ adjusts the impact of the dense loss on the discriminator's loss function, with a default value of 1.

During the training process, to ensure a strong discriminator for better guiding the generator's training, an iterative strategy is adopted. Specifically, for every five iterations of the discriminator, the generator performs one iteration. This approach enhances the discriminator's ability, thus more effectively guiding the generator's improvement.

4. Experiment and analysis

4.1. Evaluation indicators

To evaluate the model's performance, we used the following evaluation indicators: accuracy, floating point operations (FLOPs), and frames per second (FPS).

Accuracy measures the proportion of correctly classified samples out of all samples. The specific calculation method is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{7}$$

where *TP* stands for the number of correctly classified positive samples. *TN* represents the count of correctly classified negative samples. *FP* indicates the number of incorrectly classified positive samples. *FN* signifies the count of incorrectly classified negative samples.

FLOPs measure the complexity of a model by quantifying the total number of multiplication and addition operations required during model execution. A lower FLOP value indicates that the model has lower computational requirements during inference, thereby increasing the model's computational speed. $FLOPs(self\text{-}attention)$ represents the FLOPs of the self-attention mechanism, $FLOPs(feedforward)$ represents the FLOPs of the feedforward neural network, d_{ff} represents the output dimension, and N_{layers} represents the number of model layers.

$$FLOPs(self\text{-}attention) = N \times L \times (4C^2 + L \times C), \tag{8}$$

$$FLOPs(feedforward) = N \times L \times d_{ff} \times (2C + 1), \tag{9}$$

$$FLOPs = N_{layers} \times (FLOPs(feedforward) + FLOPs(self\text{-}attention)). \tag{10}$$

FPS is the number of sample images that can be classified per second during model inference.

4.2. Fine-tuning strategy

In downstream task application scenarios, fine-tuning and linear probing are two very common and effective transfer learning strategies. The core of the fine-tuning strategy is that it uses the pre-trained model as a basis and then further refines the entire network architecture to ensure that the model can better adapt to the new task requirements. This strategy adjusts all or part of the parameters of the model so that the pre-trained model can more closely fit the characteristics of the new task.

In contrast, the linear probing strategy is more concise and direct. It keeps the feature extraction part of the pre-trained model unchanged and only adds a linear classifier to the new training task. The main purpose of this strategy is to evaluate the generalization ability of pre-trained features on new tasks while reducing training time and computing resource consumption.

This paper conducts in-depth experimental analysis from three different perspectives to comprehensively evaluate the performance of these two strategies:

(1) FT means fine-tuning the model using the full dataset. This method makes full use of the prior knowledge and complete training labels of the pre-trained model and aims to fully optimize the parameter settings of the model.

(2) $FT_{30\%}$ (fine tuning with 30% data) is a variant of the FT method, which only uses 30% of the

dataset for fine-tuning. The advantage of this method is that it can evaluate the model's performance when the data is limited, which helps us better understand the generalization ability and dependence of the model on the data.

(3) $LIN_{30\%}$ (linear probing with 30% data) refers to using 30% of the data set for linear probing. In this method, the feature extraction part of the pre-trained model remains unchanged, and only a new linear classifier is trained. The primary purpose of this method is to evaluate the model's classification performance on the new task while keeping the pre-trained features unchanged.

4.3. Parameter setting

This section mainly introduces the details of the experiments. The generator uses the base version of MAE. The discriminator consists of linear layers and leaky ReLU with a 0.2 slope. The training parameters used in this experiment are shown in Table 1.

Table 1. Training parameters.

Parameter	Value
Image size	224*224
Batch size	24
Epoch	400
Generator learning rate	0.0006
Discriminator learning rate	0.0001
Optimizer	Adam
Momentum parameter	(0, 0.99)
Attenuation factor	0.99

The experiments in this paper were conducted on an Nvidia GeForce GTX 2080TI GPU and an Intel Core i7-10700K CPU @ 3.80Hz. The operating system used was Windows 10. The software environment was configured with PyTorch 1.11.0, Torchvision 0.13.1, and Timm 0.3.2. CUDA 11.3 and Python 3.7.16 were also used. The integrated development environments were PyCharm and Anaconda.

4.4. Experiment and analysis

4.4.1. Concealment rate comparison experiment

In masked autoencoders (MAE), the masking rate is crucial. This section analyzes model performance with different masking rates. Accuracy results under various masking rates are shown in Figure 6.

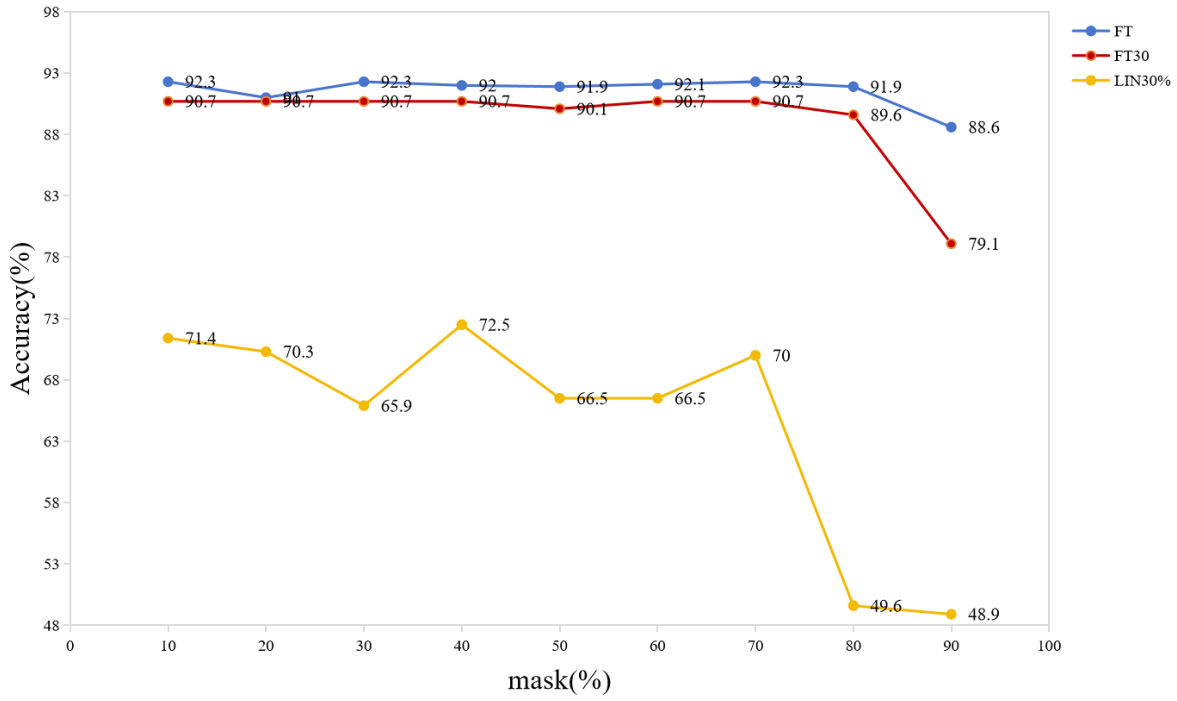


Figure 6. Comparison of accuracy under different masking rates.

Figure 6 shows that the model performs best in fine-tuning tasks with a 60% masking rate and second best with a 70% rate. For linear probing tasks, the model performs best with a 40% masking rate, similar to 70%. Considering performance in both tasks, a 70% masking rate is relatively optimal. Therefore, we set the masking rate to 70% in subsequent experiments.

4.4.2. Discriminator analysis

This section analyzes the input dimensions and normalization of the discriminator. Specific experimental details are as follows.

We compared the discriminator's input dimensions, exploring the effects of 2D vector $(\mathrm{N}, L, patch*2*3)$ versus 3D vector $(\mathrm{N}, 3, H, W)$. Table 2 shows the results for both 2D and 3D vector methods. Results indicate the 3D vector version performs slightly better than 2D. We chose the 3D vector method for subsequent experiments based on these results.

Table 2. Accuracy comparison of discriminator at different input dimensions.

	FT	$FT_{30\%}$	$LIN_{30\%}$
2D Vector	92.3	90.7	62.6
3D Vector	**92.3**	**90.7**	**70.0**

Next, we investigated whether normalization is needed in the discriminator model. Initial experiments considered adding normalization during the pre-training phase. We conducted comparative experiments, as shown in Table 3. However, results indicated that adding normalization did not improve training. Without normalization, the model's performance was significantly improved.

Table 3. Accuracy impact of regularization.

Standardization	FT	$FT_{30\%}$	$LIN_{30\%}$
√	92.2	90.1	61.0
	92.3	**90.7**	**70.0**

4.4.3. Ablation experiment

This paper verifies the effectiveness of multiple optimizations in the MAE-based GAN model through ablation experiments. Specifically, we sequentially added Wasserstein distance, gradient penalty, dense loss function, and cigarette sample image segmentation and recombination to the model, constructing several improved models. On the same test dataset, we compared the performance of these improved models with the experimental results shown in Table 4.

Table 4. Ablation experiment results.

Experiments	**Segmentation and recombination**	**dense**	**W and G**	**FT**	**$FT_{30\%}$**	**$LIN_{30\%}$**
MAE				78.9	74.2	50.5
(A)				69.3	50.5	46.1
(B)			√	80.2	78	51.2
(C)		√		79.3	77.5	42.3
(D)	√			91.8	90.1	62.7
(E)		√	√	80.3	78.6	55.5
(F)	√		√	92	90.1	65.9
(G)	√	√		90.9	88.5	63.8
(H)	√	√	√	92.3	90.7	70.0

In Table 4, experiment A uses a GAN model built from MAE as the generator and an improved discriminator. The subsequent experiments were all based on this model and further improved. Experiment B includes Wasserstein distance and gradient penalty. The results show that combining Wasserstein distance and gradient penalty can significantly improve the training effect. This method is significantly better than the method without these additions.

In experiment C, dense loss functions were added separately to the generator and discriminator. The results were not ideal. However, combining dense loss functions with Wasserstein distance and gradient penalty significantly improved training in Experiment E. Further experiments explored hyperparameter settings, showing that 1 yielded the best training results. Related results are shown in Figure 7.

Experiment D used segmentation and recombination alone. The elongated images were resized to nearly equal width and height. This method showed significant results. The experiment showed that the model's performance improved after image segmentation and recombination. Compared to models without image segmentation and recombination, the accuracy is increased by about 10%.

Experiment F used segmentation and recombination along with Wasserstein distance and gradient penalty. This improved performance compared to using either method alone. Experiment G used segmentation and recombination combined with the dense loss function.

In experiment H, the model combined segmentation and recombination, Wasserstein distance,

gradient penalty, and dense loss. The model achieved its best performance in this setup. Compared to the original MAE-based GAN, the accuracy is improved by 13% in fine-tuning with the full dataset (FT). With 30% of the dataset ($FT_{30\%}$), accuracy is improved by 16.5%. When 30% of the data are used for linear probing, the accuracy is improved by 19.5%.

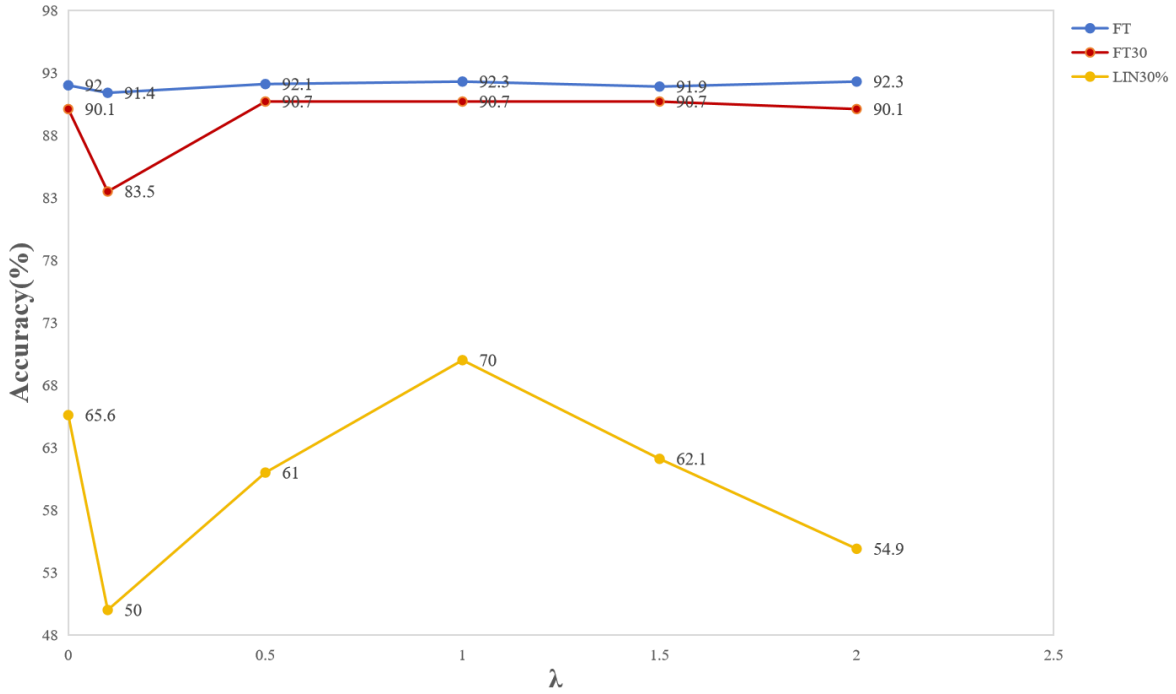


Figure 7. λ hyperparameter analysis.

4.4.4. Comparative experiments with other self-supervised learning methods

To demonstrate the effectiveness of this method, we compared it with other self-supervised learning methods. The experimental results are shown in Table 5.

Table 5. Accuracy of comparative experiments with other self-supervised learning methods.

Method	Architecture	Epochs	FT	$FT_{30\%}$	$LIN_{30\%}$	FLOPs(G)	FPS
BYOL	ResNet50	400	79.5	75.2	50.6	4.1	33.8
DINO	Small	400	79.9	74.7	51.1	4.3	35.3
IBOT	ViT-B/16	400	79.9	74.2	51.1	16.9	28.0
MOCOv3	ViT-B/16	400	79.9	74.2	51.1	16.9	28.0
MAE	ViT-B/16	400	78.9	75.3	50.5	16.9	28.0
MAE	ViT-L/16	400	77.9	74.2	59.5	59.7	24.1
Our no segmentation and recombination	ViT-B/16	400	80.3	78.6	54.9	16.9	28.0
MAE-GAN	ViT-B/16	400	92.3	90.7	70.0	16.9	28.0

In Table 5, the results show that our model outperforms others with the same architecture without image segmentation and recombination. In fine-tuning tasks using the full dataset (FT), our model

achieved higher accuracy than other models. In fine-tuning tasks using 30% of the dataset ($FT_{30\%}$), our model's accuracy was much higher than other models. In linear probing with 30% of the dataset ($LIN_{30\%}$), our model outperformed other models, except when compared to larger models. Our model's accuracy improved significantly after applying segmentation and recombination to cigarette images. FT reached 92.3%, $FT_{30\%}$ reached 90.7%, and $LIN_{30\%}$ reached 70%.

4.4.5. Comparative experiments with other image classification methods

As shown in Table 6, this model is compared with other supervised classification models. Without adding segmentation and recombination, the performance of this model is comparable to most models and even better than some, but there is still a certain gap with the best supervised classification model. However, after adding segmentation and recombination, the performance of this model is significantly improved, and it is better than other models in all aspects.

Table 6. Accuracy of comparative experiments with other image classification methods.

Method	Architecture	Epochs	Accuracy (%)	FLOPs(G)	FPS
ResNet	ResNet50	400	83.2	4.1	33.8
MoblieVit	Small	400	81.7	0.3	31.4
Transformer	ViT-B/16	400	77.9	16.9	28.0
SwinTransformer	Tiny	400	80.3	4.4	30.0
ConvNeXt	Tiny	400	82.1	4.5	36.1
MobileNetV2	-	400	82.3	0.33	35.4
EfficientnetV2	Base	400	80.7	2.9	24.2
Our no segmentation and recombination	ViT-B/16	400	80.3	16.9	28.0
MAE-GAN	ViT-B/16	400	92.3	16.9	28.0

5. Conclusions

This paper proposes a self-supervised learning-based classification model for cigarette appearance defects. This is a generative adversarial network (GAN) model based on masked autoencoders (MAE), called MAE-GAN. This study adopts the mask reconstruction method of MAE, uses MAE as the generator, and forms a GAN. The generator reconstructs the image and learns its features. This paper integrates the loss function of MAE into the loss function of GAN, so that the model focuses on pixel loss during training and improves performance. In order to stabilize the training process, this paper adds the Wasserstein distance and gradient penalty in WGAN-GP. Considering the slender characteristics of cigarette images, this paper segments the cigarette images and recombines them into nearly square images. Experiments show that the model performance significantly improves when the input image size is the same.

Of course, this study has some limitations. The discriminator is composed of several linear layers with leaky ReLU. This may not perform well with complex images or large datasets. In the future, we plan to improve the linear layers by replacing them with convolutional layers. We will also increase the depth of the discriminator to handle larger and more complex datasets. Like MAE, this model does not perform well in linear probing tasks. This is another area we plan to improve. We also attempted

to use larger versions of MAE as the generator. However, the results were unsatisfactory. We suspect the generator was too strong, which hindered GAN training. In the future, we may need a more powerful discriminator to improve model training.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was financially supported by the Yunnan Provincial Department of Science and Technology-Yunnan University Joint Special Project for Double-Class Construction, China (Grant No. 202201BF070001-005).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. G. W. Yuan, J. C. Liu, H. Y. Liu, R. Qu, H. Zhou, Cigarette appearance defect classification based on ResNeSt (Chinese), *Journal of Yunnan University (Natural Science Edition)*, **44** (2022), 464–470. https://doi.org/10.7540/j.ynu.20210257
2. R. Qu, G. W. Yuan, J. C. Liu, H. Zhou, Detection of cigarette appearance defects based on improved SSD model, *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering*, 2021, 1148–1153. https://doi.org/10.1145/3501409.3501612
3. H. Y. Liu, G. W. Yuan, Cigarette appearance defect detection method based on improved YOLOv5s, *Comput. Technol. Dev.*, **32** (2022), 161–167.
4. G. W. Yuan, J. C. Liu, H. Y. Liu, Y. H. Ma, H. Wu, H. Zhou, Detection of cigarette appearance defects based on improved YOLOv4, *Electron. Res. Arch.*, **31** (2023), 1344–1364. https://doi.org/10.3934/era.2023069
5. H. Y. Liu, G. W. Yuan, L. Yang, K. X. Liu, H. Zhou, An appearance defect detection method for cigarettes based on C-CenterNet, *Electronic*, **11** (2022), 2182–2182. https://doi.org/10.3390/electronics11142182
6. Y. H. Ma, G. W. Yuan, K. Yue, H. Zhou, CJS-YOLOv5n: a high-performance detection model for cigarette appearance defects, *Math. Biosci. Eng.*, **20** (2023), 17886–17904. https://doi.org/10.3934/mbe.2023795
7. Q. Z. Xie, Z. H. Dai, E. Hovy, M. Luong, Q. Le, Unsupervised data augmentation for consistency training, *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, 6256–6268.

8. J. Z. Bengar, J. van de Weijer, B. Twardowski, B. Raducanu, Reducing label effort: self-supervised meets active learning, *Proceedings of IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, 1631–1639. https://doi.org/10.1109/ICCVW54120.2021.00188
9. X. Liu, F. J. Zhang, Z. Y. Hou, L. Mian, Z. Y. Wang, J. Zhang, et al., Self-supervised learning: generative or contrastive, *IEEE Trans. Knowl. Data Eng.*, **35** (2023), 857–876. https://doi.org/10.1109/TKDE.2021.3090866
10. S. Kornblith, J. Shlens, Q. V. Le, Do better ImageNet models transfer better? *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, 2661–2671. https://doi.org/10.1109/CVPR.2019.00277
11. Y. W. Li, J. J. Qiao, S. J. Ma, Z. Q. Wu, Q. H. Wu, Research and improvement of cigarette splicing quality of cigarette rolling machine (Chinese), *Tobacco Science and Technology*, **45** (2012), 24–27. https://doi.org/10.3969/j.issn.1002-0861.2012.10.006
12. L. L. Jing, Y. L. Tian, Self-supervised visual feature learning with deep neural networks: a survey, *IEEE Trans. Pattern Anal.*, **43** (2020), 4037–4058. https://doi.org/10.1109/TPAMI.2020.2992393
13. K. M. He, X. L. Chen, S. N. Xie, Y. H. Li, P. Dollr, R. Girshick, Masked autoencoders are scalable vision learners, *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, 16000–16009. https://doi.org/10.1109/CVPR52688.2022.01553
14. Y. Li, S. Xie, X. Chen, P. Dollar, K. He, R. Girshick, Benchmarking detection transfer learning with vision transformers, arXiv: 2111.11429. https://doi.org/10.48550/arXiv.2111.11429
15. K. M. He, H. Q. Fan, Y. X. Wu, S. N. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 9729–9738. https://doi.org/10.1109/CVPR42600.2020.00975
16. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial nets, *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, 2672–2680.
17. A. Radford, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv: 1511.06434. https://doi.org/10.48550/arXiv.1511.06434
18. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved tech- niques for training gans, *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, 2234–2242.
19. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, *International Conference on Machine Learning*, 2017, 214–223.
20. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved training of Wasserstein GANs, *Proceedings of the 34th International Conference on Machine Learning*, 2017, 214–223.
21. T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, arXiv: 1710.10196. https://doi.org/10.48550/arxiv.1710.10196

AIMS Press