JEI Journal of Electronic Imaging

RESEARCH PAPER

Multidomain feature fusion method for small object classification: MDFF

Jing Hu,[a] Zican Shi,[b,*] Zheng Zhang,[b] Siqi Lv,[b] Yifan Chen,[b] Yan Ouyang,[c] and Jia He[d]

[a]National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, Wuhan, China
[b]Huazhong University of Science and Technology, Wuhan, China
[c]Air Force Early Warning Academy, Wuhan, China
[d]Chinese People's Liberation Army 95841 Troops, China

ABSTRACT. The task of classifying small objects is still challenging for current deep learning classification models [such as convolutional neural networks (CNNs) and vision transformers (ViTs)]. We believe that these algorithms are not designed specifically for small targets, so their feature extraction abilities for small targets are insufficient. To improve the classification capabilities of CNN-based and ViT-based classification models for small objects, two multidomain feature fusion (MDFF) frameworks are proposed to increase the amount of feature information derived from images and they are called MDFF-ConvMixer and MDFF-ViT. Compared with the basic model, the uniquely added design includes frequency domain feature extraction and MDFF processes. In the frequency domain feature extraction part, the input image is first transformed into a frequency domain form through discrete cosine transform (DCT) transformation and then a three-dimensional matrix containing the frequency domain information is obtained via channel splicing and reshaping. In the MDFF part, MDFF-ConvMixer splices the spatial and frequency domain features by channel, whereas MDFF-ViT uses a cross-attention mechanism to fuse the spatial and frequency domain features. When targeting small target classification tasks, these two frameworks obviously improve the utilized classification algorithm. On the DOTA dataset and the CIFAR10 dataset with two downsampling operations, the accuracies of MDFF-ConvMixer relative to ConvMixer increase from 87.82% and 62.14% to 90.14% and 66.00%, respectively, and the accuracies of MDFF-ViT relative to the ViT increase from 79.22% and 36.2% to 88.15% and 59.23%, respectively.

Keywords: ConvMixer; vision transformer; small object classification; frequency domain

Paper 230360G received Mar. 29, 2023; revised Jun. 3, 2023; accepted Jun. 23, 2023; published Jul. 10, 2023.

1 Introduction

Small target recognition technology is very important in the fields of medical image analysis, security systems, video surveillance and tracking, automatic driving, etc. Regardless of whether it involves traditional machine learning methods or recognition algorithms based on deep learning; however, the classification effect achieved for small targets is still not satisfactory.[1] Compared with regular-sized targets, small targets usually have only dozens of pixels or a few pixels and present problems, such as low resolution and a lack of image information, so the feature expression abilities of small targets are weak.[2] Deep learning methods have achieved

*Address all correspondence to Zican Shi, 741424985@qq.com

impressive results on regular-sized targets.[3] Convolutional neural networks (CNNs) generally use methods such as network deepening and feature multiplexing to enhance the classifier's ability to extract spatial features from the target.[4,5] ResNet[6] uses the idea of residual learning. On the basis of VGG19,[7] a residual unit is added through a shortcut to solve the degradation problem of the deep network so that the network becomes deeper and can extract deeper features. However, deeper networks are more likely to lead to the loss of small target features. Although DenseNet[8] establishes dense connections between different layers, reuses features between the front and rear layers, and performs well on large targets, because small targets carry less information, it is easy to cause overfitting by directly using DenseNet to deepen the network. Xu et al.[9] proposed a new target feature extraction approach, which uses adaptive channel pruning to reshape images in the frequency domain and then uses conventional CNNs for classification. This method uses the frequency domain features of the target and its final effect is better than that of the original CNN method. However, simply using frequency domain features and pruning leads to incomplete feature extraction for small objects, so this method is still not suitable for small object classification. Through the learning processes of the above methods and our understanding of small target features, we believe that the feature extraction ability of a model for small targets can be enhanced by introducing a combination of frequency domain features and spatial domain features. Therefore, a spatial and frequency domain feature fusion method [multidomain feature fusion (MDFF)] based on data enhancement is proposed in this paper, which enhances the importance of the frequency domain features to the classifier, enriches the effective features of small targets, and improves the model's small target recognition ability. Based on this method, two recognition frameworks [MDFF-ConvMixer and MDFF-vision transformer (ViT)] are designed. On the DOTA dataset and the CIFAR10 dataset with two downsampling operations, we verify that the classifiers constructed by these two frameworks achieve improved recognition performance for small objects. For the small target classification task, this paper is a new attempt to fuse spatial and frequency domain features.

2 Related Works

The main work of this paper is to carry out research on classification and recognition technology for small targets. The research idea is to realize the extraction and fusion of the spatial and frequency domain features of targets based on the structures of ConvMixer and a vision transformer (ViT). In this chapter, we focus on some related work.

2.1 ViTs

In recent years, due to the success of transformers in the field of natural language processing (NLP), nonconvolution models that only rely on transformers have gradually become the most advanced algorithms in the field of computer vision. The transformer structure with attention proposed by Vaswani et al.[10] has achieved good results. The bidirectional encoder representations from transformers (BERT) method proposed by Devlin et al.[11] uses a classification (CLS) token to aggregate the classification information of the entire token to reduce the computational complexity of the transformation algorithm. Later, a self-attention technique was proposed by Parmar et al.[12] This method focuses only on the local neighborhoods of pixels, which enables the application of transformers in vision tasks. The ViT[13] borrows from previous work and encodes an image into several tokens with location information; this was the first approach to match or even surpass CNNs in terms of performance on vision tasks with a transformer-based algorithm. Cross ViT[14] utilizes a dual-path structure to improve the effect of the ViT on multiscale features. Based on the above algorithms, we propose a MDFF method for feature augmentation to better utilize ViTs for the visual representation of small objects.

2.2 CNNs

Although ViTs can outperform CNNs on some tasks with additional pretraining on big data, the lower dataset requirements and faster training speeds of CNNs make them some of the dominant frameworks for vision tasks. A large number of scholars have conducted in-depth research on CNNs. For example, ResNet[6] uses the idea of residual learning to deepen convolutional networks, and DenseNet[8] further reduces the number of required model parameters by reusing

features. These improvements all provide CNNs with more powerful classification capabilities. ConvMixer[15] was inspired by the ViT. It directly operates on patches and performs the convolution operation, which further improves the classification ability of the convolutional model. However, the above methods are not suitable for small object classification because when the object sizes are too small, deeper networks with more spatial convolutions tend to lose features more easily. Based on the above considerations, in this paper, the frequency domain features are embedded in a ConvMixer-based MDFF framework to achieve feature enhancement.

2.3 Frequency Domain Feature Extraction

The frequency domain representations of images contain rich information and making full use of this information can improve a computer's understanding of various image processing tasks. Hsu et al.[16] was the first to use the Mandala transform to identify targets. Shen and Sethi[17] directly extracted low-level frequency domain features from images to detect regions of interest and edges. Both Ehrlich and Davis[18] and Gueguen et al.[19] skipped the JPEG decoding step and directly used frequency domain information for learning. Ehrlich and Davis[18] proposed a general learning algorithm in the JPEG transform domain for interconversion between spatial and frequency domain networks, whereas Gueguen et al.[19] used an intermediate JPEG codec module to extract frequency domain features to train a CNN model for image classification. Gueguen et al.[19] also considered DCT to be an alternative convolution. Xu et al.[9] analyzed spectral bias from the perspective of the frequency domain, proposed a learning-based channel pruning algorithm to prune frequency components that are of little use and used frequency domain information as the input of commonly used neural networks. These methods only consider the frequency domain features and do not consider the fusion of frequency domain features and spatial domain features. The work in this paper is an attempt to fuse the spatial and frequency domain features.

3 Methodology

Our feature fusion method is designed on ConvMixer and ViT models for small object classification. Therefore, this chapter first briefly introduces the ConvMixer and ViT models and then describes our proposed algorithm frameworks (MDFF-ConvMixer and MDFF-ViT) in detail.

3.1 ViT and ConvMixer Frameworks

A ViT splits an entire image into small image patches and then converts these small patches into linear embedding sequences via linear projection.[13] Since this splitting process loses the position information of the image block, which is indispensable in vision tasks, the ViT adds a position embedding to each token. Similar to BERT's CLS token, an additional CLS token is added to the front of each sequence to facilitate the final classification step. All tokens of a sequence are fed into multiple transformer encoders as inputs, but the final classification process uses only the CLS tokens, not all tokens, because after multiple encoding iterations, the CLS tokens already contain important information from other tokens. A transformer encoder consists of multiple stacked blocks, each of which consists of multiheaded self-attention[12] and a multilayer perceptron (MLP).[20] It is worth noting that since CLS tokens can contain rich feature information, we try to achieve joint feature enhancement by exchanging the CLS tokens and branch tokens of different domain features.

ConvMixer consists of a patch embedding block and multiple repeated fully convolutional blocks. Although ConvMixer's patch embedding block is similar to the ViT's linear projection function, it is implemented through a 2D convolution operation. Each fully convolutional block of ConvMixer consists of grouped convolution (the number of groups equals the number of channels) and point convolution (the convolution kernel has a size of 1×1). The group-convolved feature map and the point-convolved feature map undergo residual learning. A pooling and normalization layer is located after each convolution operation to reduce the computational cost of the model. Although Trockman and Kolter[15] believed that ConvMixer is similar to the ViT in terms of its idea, its architecture is more similar to those of CNNs, such as ResNet. Due to the superior performance of ConvMixer on small target recognition tasks, this paper uses the

ConvMixer model to improve the proposed approach. While adding frequency domain features, we design a feature fusion recognition framework based on ConvMixer by performing cross-domain feature splicing according to the channel dimension.

Aiming at the difficulty of small target feature extraction, we introduce frequency domain features and use MDFF to enhance the classification ability of the model. This design idea broadens the feature extraction channels, rather than only mining features based on depth, so it is more suitable for the extraction of small target features.

3.2 Frequency Domain Feature Extraction

For small targets with insufficient spatial information, common classifiers do not perform well. For example, CNNs and ViTs, similar to the human eye, tend to pay more attention to information such as the textures and positions of images when performing classification, which are all spatial features. For objects with higher resolutions and larger sizes, a classifier can achieve better classification results by using only spatial features. For small targets, the advantages of these classifiers, such as their deeper networks and extra spatial convolutions, do not improve the classification effect but rather interfere with the extraction of identifiable features. For this reason, we propose an MDFF method that attempts to increase the frequency domain features to achieve feature enhancement, which is accomplished by expanding the feature domain rather than adding depth features. The frequency domain feature extraction method for the deep learning network is shown in Fig. 1.

As shown in the Fig. 1, the resized and cropped RGB image is denoted as x_{spa}, and the image x^2_{spa} is obtained by performing upsampling twice using the bilinear interpolation method. x^2_{spa} obtains a one-dimensional matrix of the Y channel ($Y\_x^2_{\text{spa}}$) through DCT transformation, and x_{spa} obtains one-dimensional Cb and Cr matrices ($Cb\_x^2_{\text{spa}}$, $Cr\_x^2_{\text{spa}}$) through DCT transformation. When performing DCT transformation, the input image is divided into multiple 8*8 matrices, DCT transformation is performed on each matrix, and a frequency coefficient matrix is obtained after the transformation. The two-dimensional DCT coefficients with the same frequency are divided into the same group to form a channel. Later, reshaping and concatenation are used to deform the Y, Cb, and Cr matrices into tractable forms for normalization. Finally, the matrices are reshaped into a three-channel image, denoted as x_{fre}, to facilitate the subsequent feature extraction and fusion processes.

3.3 Spatial and Frequency Domain Feature Fusion

To study the small target recognition effect attained after adding frequency domain features, we improve the feature fusion abilities of two different recognition models, including the state-of-the-art CNN-based ConvMixer model and the classic transform-based ViT classification model. We refer to the improved recognition frameworks as MDFF-ConvMixer and MDFF-ViT, respectively.

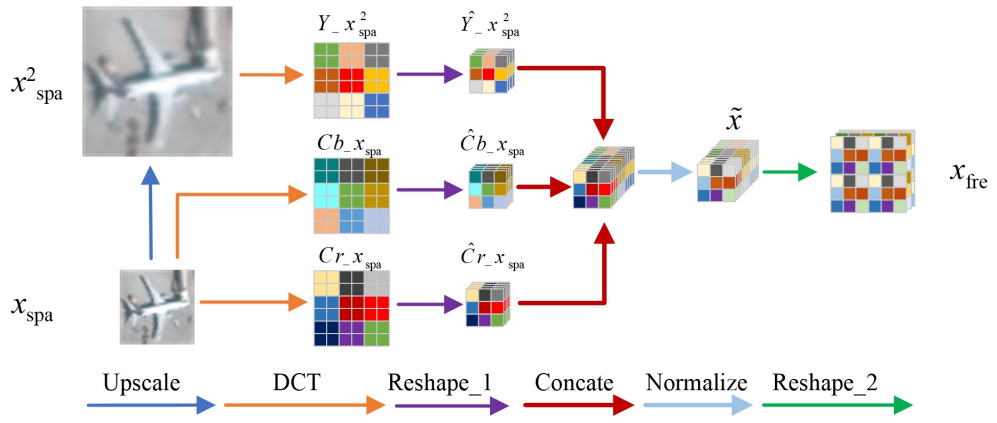

Fig. 1 Frequency domain feature extraction.

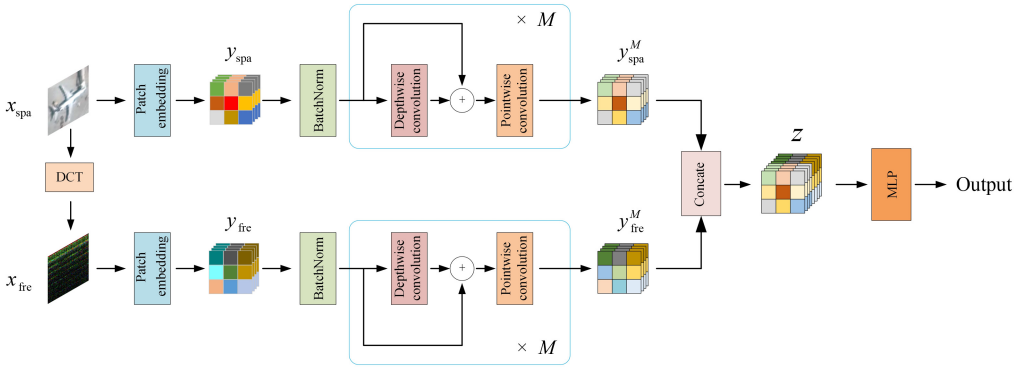

Fig. 2 MDFF-ConvMixer flow chart.

3.3.1 *MDFF-ConvMixer*

As shown in Fig. 2, the input image is subjected to simple image preprocessing techniques (such as resizing, cropping, and rotation) and frequency domain feature extraction (in Sec. 3.2) to obtain the target spatial image $\in \mathbb{R}^{c\times n\times n}$ and the frequency domain image $x_{\text{fre}} \in \mathbb{R}^{c\times n\times n}$. x_{spa} and x_{fre} obtain the spatial feature block $y_{\text{spa}} \in \mathbb{R}^{h\times(\frac{n}{p})\times(\frac{n}{p})}$ and the frequency domain feature block $y_{\text{fre}} \in \mathbb{R}^{h\times(\frac{n}{p})\times(\frac{n}{p})}$, respectively, through their patch embedding modules. Each patch embedding module here is composed of a convolutional layer, an activation function and a normalization layer.[15] The convolutional layer's kernel size $=p_1$, the stride $=p_1$, and p_1 is the patch size. The transformation process is shown in the following equations:

$$y_{\text{spa}} = \text{BN}(\sigma\{\text{Conv}_{c_{\text{in}}\to h}(x_{\text{spa}})\})|, \tag{1}$$

$$y_{\text{fre}} = \text{BN}(\sigma\{\text{Conv}_{c_{\text{in}}\to h}(x_{\text{fre}})\}). \tag{2}$$

The feature blocks obtained by y_{spa} and y_{fre} through overlapping ConvMixer layers are denoted as $y_{\text{spa}}^{m} \in \mathbb{R}^{h\times(\frac{n}{p})\times(\frac{n}{p})}$ and $y_{\text{fre}}^{m} \in \mathbb{R}^{h\times(\frac{n}{p})\times(\frac{n}{p})}$, respectively, where m is the number of overlapping ConvMixer layers. Each ConvMixer layer is composed of depthwise convolution and pointwise convolution.[15] Depthwise convolution is actually a grouped convolution with the number of groups equal to the number of channels, whereas pointwise convolution is a 1×1 point convolution. An activation function and a normalization layer are located after each convolution. Before and after performing depthwise convolution, the features are connected by their residuals,[15] as shown in Eqs. (3)–(6). The depthwise convolution and pointwise convolution operations are shown in Eqs. (7) and (8), respectively

$$\hat{y}_{\text{spa}}^{t} = \text{BN}(\sigma\{\text{DepConv}_{h\to h}(y_{\text{spa}}^{t-1})\}) + y_{\text{spa}}^{t-1}, \tag{3}$$

$$\hat{y}_{\text{spa}}^{t} = \text{BN}(\sigma\{\text{PotConv}_{h\to h}(\hat{y}_{\text{spa}}^{t})\}), \tag{4}$$

$$\hat{y}_{\text{fre}}^{t} = \text{BN}(\sigma\{\text{DepConv}_{h\to h}(y_{\text{fre}}^{t-1})\}) + y_{\text{fre}}^{t-1}, \tag{5}$$

$$\hat{y}_{\text{fre}}^{t} = \text{BN}(\sigma\{\text{PotConv}_{h\to h}(\hat{y}_{fre}^{t})\}), \tag{6}$$

$$\text{DepConv} = \text{Conv}(\text{stride} = p_2, \text{kernel} = p_3), \tag{7}$$

$$\text{PotConv} = \text{Conv}(\text{kernel} = 1). \tag{8}$$

Among them, the initial values of y_{spa}^{t-1} and y_{fre}^{t-1} are y_{spa} and y_{fre}, respectively; t is the variable representing the number of stacking iterations; and the value range is $1 \le t \le \text{M}$. p_2 represents the stride parameter in the depthwise convolution, which is determined by the sizes of the input and output. The function of p_2 is to ensure that the size of each feature remains unchanged before and after the convolution operation.[21] p_3 represents the kernel size in the depthwise convolution, which is generally 7. The kernel size of point convolution is set to 1.

After obtaining the spatial domain depth feature y_{spa}^{M} and the frequency domain depth feature y_{fre}^{M}, the fusion feature $z \in \mathbb{R}^{(2*h)\times(\frac{n}{p})\times(\frac{n}{p})}$ is obtained through channel splicing, as Eq. (11) shows

$$z = [y_{\text{spa}}^{M} \| y_{\text{fre}}^{M}]. \tag{9}$$

The feature z can yield the output category result after going through the fully connected layer. And the loss function of MDFF-ConvMixer is calculated using cross-entropy and is defined as

$$\text{Loss} = -\sum_{c=1}^{M} q^{ic} \log(p^{ic}), \tag{10}$$

where M represents the number of categories. q^{ic} is an sign function that takes a value of 0 or 1. If the true category of sample i is c, it takes the value of 1; otherwise, it takes the value of 0. The probability p^{ic} represents the likelihood of sample i belonging to category c, which is obtained by inputting the feature z into a fully connected layer.

To study the best location for fusing the spatial and frequency domain features, we make attempts with different strategies, such as the following.

MC-strategy1: fuse x_{spa} and x_{fre} before extracting features;

MC-strategy2: fuse image patches y_{spa} and y_{fre} before extracting features;

MC-strategy3: use attention mechanism to perform feature fusion on y_{spa}^{M} and y_{fre}^{M} after feature extraction;

MC-strategy4: plus the last two outputs of spa-branch and fre-branch instead of fusing features, which means combining $\text{output}_{\text{spa}}$ and $\text{output}_{\text{fre}}$ to get a new output, expressed in the formula as: $\text{output}_{\text{new}} = \text{output}_{\text{spa}} + \text{output}_{\text{fre}}$;

MC-strategy5: take the element-wise maximum of the outputs from the spa-branch and fre-branch to get a new output: $\text{output}_{\text{new}} = \max(\text{output}_{\text{spa}}, \text{output}_{\text{fre}})$.

Through comparative experiments, we find that the optimal feature fusion method for the ConvMixer model performs feature splicing before the fully connected layer. We believe that this is because the fusion mechanisms of MC-strategy1 to MC-strategy3 may destroy the location information contained in airspace features, and MC-strategy4 to MC-strategy5 fail because the meanings of loss in the airspace and frequency domains are quite different and their loss are difficult to fuse. Detailed ablation experiments can be found in Sec. 4.3.

3.3.2 *MDFF-ViT*

As shown in Fig. 3, the input image is subjected to simple image preprocessing technology to obtain the spatial domain image x_{spa}, and then the frequency domain image x_{fre} is obtained through frequency domain feature extraction (shown in Sec. 3.2). Linear projection is used to process x_{spa} and x_{fre} to obtain two different tokens (T_{spa} and T_{fre}, respectively). T_{spa} and T_{fre} are processed through the token fusion module to obtain two fusion features $T_{\text{fre+spa}}$ and $T_{\text{spa+fre}}$, respectively. It should be noted that the input of the token fusion module includes two tokens, and the output also contains two tokens. A token can be split into a CLS token and multiple patch tokens, where the CLS token contains most of the information of the entire token. $T_{\text{fre+spa}}^{\text{cls}}$ contains most of the information in the fusion feature $T_{\text{fre+spa}}$, and $T_{\text{spa+fre}}^{\text{cls}}$ contains most of the information in the fusion feature $T_{\text{spa+fre}}$. Our processing approach sends $T_{\text{fre+spa}}^{\text{cls}}$ and $T_{\text{spa+fre}}^{\text{cls}}$ to 2 separate MLP heads and finally performs linear fusion on the two losses.

Effective feature fusion is the key to learning multidomain feature representations. After testing several strategies, such as self-attention feature fusion, simple token splicing and fusion, etc. Fusion scheme details can be found in Sec. 4.3. We choose the token fusion module based on the cross-attention mechanism and its design idea is inspired by Cross ViT.[14] Each token fusion module in MDFF-ViT consists of two parallel transformer encoders and a cross-attention

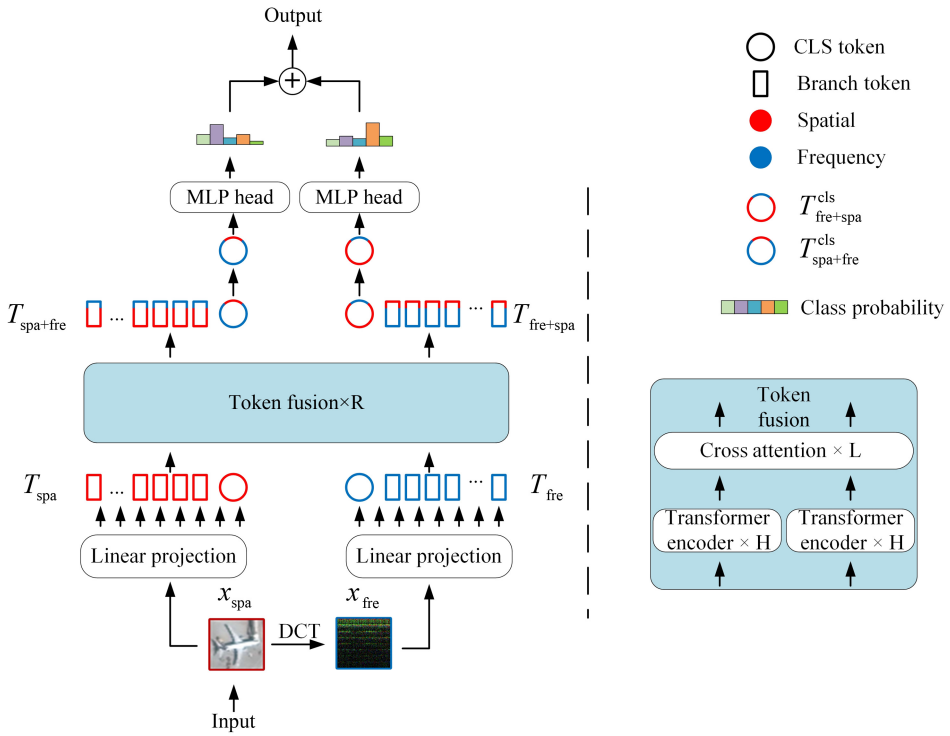

Fig. 3 MDFF-ViT flow chart.

mechanism. The two transformer encoders separately extract the spatial and frequency domain features of the data, and the attention mechanism is mainly responsible for processing the feature fusion part. The features obtained after the transformer encoder are denoted as $\hat{T}_{fre}$ and $\hat{T}_{spa}$, respectively, and they are cross-fused using the cross-attention mechanism. The following describes the process of feature cross fusion, as shown in Eqs. (11)–(15). Taking $\hat{T}_{fre}$ fusing the information of $\hat{T}_{spa}$ to obtain $\hat{T}_{fre+spa}$ as an example, $\hat{T}_{fre}$ can be split into $\hat{T}^{cls}_{fre}$ and $\hat{T}^{branch}_{fre}$, and $\hat{T}_{spa}$ can be split into $\hat{T}^{cls}_{spa}$ and $\hat{T}^{branch}_{spa}$:

$$\hat{T}_{fre} = [\hat{T}^{cls}_{fre} | \hat{T}^{branch}_{fre}], \quad \hat{T}^{b}_{spa} = [\hat{T}^{cls}_{spa} | \hat{T}^{branch}_{spa}]. \tag{11}$$

When calculating the QKV matrix, cross-attention can utilize different processing methods.[22] QKV matrix represents the query matrix, key matrix and value matrix proposed in Ref. 10. As shown in Eq. (12), q is calculated through the linear projection of $\hat{T}^{cls}_{fre}$, k is calculated by the simple concatenation of the linear projection results of $\hat{T}^{cls}_{fre}$ and $\hat{T}^{branch}_{spa}$, and the calculation process of v is similar to that of k. The purpose of linear projection is to align the dimensions of $\hat{T}^{cls}_{fre}$ and $\hat{T}^{branch}_{spa}$ to facilitate subsequent feature cross-fusion calculations. Linear projection is achieved by adding several linear layers, and the linear projections in the spatial and temporal domains are represented by the functions $f_{spa}(\cdot)$ and $f_{fre}(\cdot)$, respectively. The calculation equations of q, k, and v are as follows:

$$q = f_{fre}(\hat{T}^{cls}_{fre}) W_q, \tag{12}$$

$$k = [f_{fre}(\hat{T}^{cls}_{fre}) \| \hat{T}^{branch}_{spa}] W_k, \tag{13}$$

$$v = [f_{fre}(\hat{T}^{cls}_{fre}) \| \hat{T}^{branch}_{spa}] W_v. \tag{14}$$

Among them, W_q, W_k, and W_v are learnable parameters.

After calculating the QKV matrix, the CLS token $\hat{T}^{\text{cls}}_{\text{fre+spa}}$ fused with spatial information can be obtained. $\hat{T}^{\text{cls}}_{\text{fre+spa}}$ is aligned and spliced with $\hat{T}^{\text{branch}}_{\text{fre}}$ through the backprojection function $g_{\text{fre}}(\cdot)$ to obtain the fused token, which is denoted as $T_{\text{fre+spa}}$. This is shown in the following equation:

$$T_{\text{fre+spa}} = [g_{\text{fre}}(\hat{T}^{\text{cls}}_{\text{fre+spa}}) \| T^{\text{branch}}_{\text{fre}}]. \tag{15}$$

The fusion of T_{spa} and T_{fre} is also a similar process. For better feature extraction and feature fusion, the token fusion module needs to overlap R times.

Here, the feature cross-fusion process of MDFF-ViT is more complicated than that of MDFF-ConvMixer. MDFF-ViT uses cross-attention for feature fusion, whereas MDFF-ConvMixer uses simple channel concatenation for fusion. The reason for this design in this paper is that the tokens in MDFF-ViT have location information, and the frequency domain features do not destroy the location information of the spatial domain features during feature crossover; thus, the fusion process is more sufficient. Subsequent experiments demonstrate that sufficient feature fusion can achieve higher performance gains.

It should be noted that in the entire MDFF-ViT framework, the process leading up to the sum of MLP heads is coherent. However, unlike the concatenation method employed in MDFF-ConvMixer to merge spatial and frequency-domain features, we adopt a different approach in MDFF-ViT. Instead, MDFF-ViT feeds the fusion results of the two feature layers ($T^{\text{cls}}_{\text{fre+spa}}$, $T^{\text{cls}}_{\text{spa+fre}}$) into two MLP layers separately. The MLP heads output two recognition scores derived from the feature mappings, and the final fusion decision is obtained by summing the two recognition scores. The loss function of MDFF-ViT is similar to that of MDFF-ConvMixer, both utilizing cross-entropy for computation. However, due to the presence of an additional MLP head in MDFF-ViT, the probability score calculation is diffient. The loss function of MDFF-ViT is expressed as follows:

$$\text{Loss} = -\sum_{c=1}^{N} q_{ic} \log(p^{ic}_{\text{spa+fre}} + p^{ic}_{\text{fre+spa}}), \tag{16}$$

where N represents the number of categories, and q^{ic} denotes the sign function, taking binary values of 0 or 1. Specifically, q^{ic} takes the value of 1 if sample i belongs to the true category c, otherwise it is set to 0. The probabilities $p^{ic}_{\text{spa+fre}}$ and $p^{ic}_{\text{fre+spa}}$ represent the likelihood of sample i belonging to category c. These two probabilities are obtained by passing the features $T^{\text{cls}}_{\text{spa+fre}}$ and $T^{\text{cls}}_{\text{fre+spa}}$ through two MLP heads separately.

3.4 Enhancement Analysis of Feature Representation

Our framework is not limited to the improvement of some special spatial models such as ConvMixer and ViT. It can also be applied to enhance other spatial feature models easily, including recognition methods based on contour features, such as (Misra, 2018),[23] (Asem, 2018),[24] and (Saleem, 2019).[25] Due to the significant differences between the spatial and frequency domain features of small object, our improvement strategy can remarkably enhance the information content of the original features (spatial) by introducing frequency domain features. This naturally leads to improved recognition performance.

The feature maps before and after feature fusion are shown in Fig. 4. x_{spa}, y^{M}_{spa}, y^{M}_{fre}, and Z in Fig. 4 represent the input RGB image, the spatial feature map, the frequency feature map, and the fused feature map, respectively. For better visualization, all of them have been scaled to a size of 256×256. It can be intuitively observed that the frequency domain features enrich the information content of the spatial domain features, which is the key to our model's excellent performance.

4 Experiments

In this section, we present experiments and their results to demonstrate the effectiveness of our proposed method.

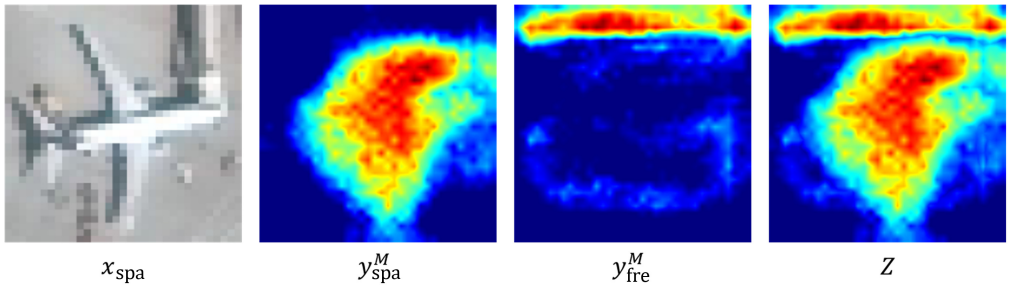

Fig. 4 Visualization feature maps of spatial, frequency, and fused features.

4.1 Experiments Setup

4.1.1 *Dataset*

Our purpose is to conduct research on recognition technology for small objects with sizes between 8*8 and 32*32. Due to the lack of such publicly available datasets, we use the downsampled DOTA dataset. The target areas marked by the DOTA dataset are cropped and downsampled to 1/4 of the original images, and the targets with pixel areas less than 32*32 are retained as the classified dataset. The dataset composition is shown in Fig. 5. In the following, the original dataset is recorded as $\text{Dota}_{32\times32}$, and the smaller target dataset obtained after continued downsampling of $\text{Dota}_{32\times32}$ is recorded as $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$. These two datasets are subsequently used for experiments to test the performance of the algorithm on small targets with different sizes. The ratio of the training set to the test set is ~5:3. $\text{Dota}_{32\times32}$ has 6 types of positive samples and 1 type of negative sample, for a total of 15,065 samples, and $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$ has the same breakdown. The training sets of $\text{Dota}_{32\times32}$ and $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$ are shown in the following figure. We also utilized the publicly available dataset cifar10[26] and downsampled it by 1/2 to meet the requirements of our research, referred to as $\text{cifar10}_{\frac{32}{2}\times\frac{32}{2}}$. In addition to $\text{Dota}_{32\times32}$, $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$, and $\text{cifar10}_{\frac{32}{2}\times\frac{32}{2}}$, we conducted experiments on the publicly available dataset Fashion-MNIST.[27]

4.1.2 *Training and evaluation*

When conducting control experiments, we set the same hyperparameters for the same set of experiments. We run the experiments for 200 epochs on 2 pieces of 3080Ti GPUs. The optimizer uses adaptive moment estimation (Adam), the default batch size is set to 64 (dynamically adjusted to models), the initial learning rate is set to 0.0001, the learning rate decay coefficient is 0.9, and the number of learning rate decay iterations is 20. The datasets are resized to 256 before being input into the classifier, and simple data enhancements such as flipping and

Category	Plane	Harbor	Small-vehicle	Storage-tank	Swimming-pool	Tennis-court	Unknown
Numbers	2758	2134	3081	2320	1371	1269	2132

Fig. 5 Information of $\text{Dota}_{32\times32}$ and $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$ and picture examples of various samples.

cropping are used. The convolutional models participating in the experimental comparison include ResNet50, Desnet, HorNet,[28] and ConvMixer, and the transformer models include Deit,[29] ViT-pre, ViT, Cross ViT-pre, Cross ViT, Swin-Transformer,[30] and CSWin-Transformer.[31] Among them, the patch embeddings of ViT-pre, ViT, Cross ViT-pre, Cross ViT, and MDFF-ViT are all linear, the patch sizes of the ViT and MDFF-ViT are 16,[13] the small patch size of Cross ViT is 16, and the large patch size is 64.[14] In the experiment, Deit's teacher model is ResNet50, and the patch size is 16.[29] ConvMixer[15] and MDFF-ConvMixer have 256 dimensions and depths of 24. We use the first accuracy attained on the test set as a model performance evaluation metric.

4.1.3 *Training process*

The training processes of MDFF ConvMixer and MDFF ViT are largely similar to the original algorithms, both following an end-to-end approach. The only difference is that original ConvMixer and ViT only require spatial domain images and labels as inputs during training, whereas MDFF ConvMixer and MDFF ViT additionally require frequency domain images. The frequency domain images are generated together with the dataloader when the dataset is loaded, by using CPU. Therefore, our conversion method does not introduce any additional GPU time. Moreover, the conversion of spatial domain images to frequency domain images by the CPU is also fast, taking less than 1 min to convert 100,000 images to frequency domain images on an Intel Xeon Gold 6330. However, our MDFF method also has its limitations. Due to the addition of frequency domain images, the memory usage and the GPU training time of the new network also nearly doubles compared to the original network.

4.2 Main Results

The experimental results are shown in Table 1. Except for the models with "-pre" after their name, the models appearing in this paper do not load any pre-trained weights. The ViT,

Table 1 Comparisons with transformer models and convolutional models on $Dota_{32\times32}$, $Dota_{\frac{32}{2}\times\frac{32}{2}}$ and $cifar10_{\frac{32}{2}\times\frac{32}{2}}$ and Fashion-MNIST.

Models	Params (M)	FLOPs (G)	$Dota_{32\times32}$ (%)	$Dota_{\frac{32}{2}\times\frac{32}{2}}$ (%)	$cifar10_{\frac{32}{2}\times\frac{32}{2}}$ (%)	Fashion MNIST (%)
			Convolutional			
ResNet50 (2016)	25.55	4.13	74.13	67.24	62.26	94.97
DenseNet (2017)	7.97	4.39	79.65	73.30	59.76	94.86
HorNet-gf (2022)	49.64	11.36	87.56	82.47	***63.73***	95.01
ConvMixer (2022)	1.95	2.54	***87.82***	***83.48***	62.14	***95.28***
MDFF-ConvMixer (ours)	3.90	5.09	**90.14**	**84.77**	**66.00**	**95.55**
			Transformer			
Deit (2021)	62.40	47.82	85.00	78.68	53.27	91.47
ViT-pre (2020)	99.00	53.36	***88.13***	**83.61**	58.25	**95.06**
ViT (2020)	99.00	53.36	79.22	78.62	36.20	90.63
Cross ViT-pre (2021)	50.54	3.50	86.61	79.41	56.81	92.30
Cross ViT (2021)	50.54	3.50	86.53	79.14	55.84	92.10
SwinV1 (2021)	48.77	11.14	85.74	82.73	47.41	94.31
CSWin (2022)	34.14	8.38	87.62	***83.14***	***58.65***	94.36
MDFF-ViT (ours)	38.62	7.96	**88.15**	79.75	**59.23**	***94.53***

Note: Bold values represent the best result, and bold italic values represent the suboptimal values.

Deit, and Cross ViT are commonly used transformer models, and ResNet and DenseNet are commonly used convolutional models.

4.2.1 *Convolutional models*

It can be seen from Table 1 that in the comparison among the convolutional models, MDFF-ConvMixer with the MDFF module achieves the best top-1 accuracy indices on $\text{Dota}_{32\times32}$, $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$, $\text{cifar}_{\frac{32}{2}\times\frac{32}{2}}$, and Fashion-MNIST. MDFF-ConvMixer is an extension of the ConvMixer model that incorporates a frequency-domain branch, resulting in twice the parameter count and FLOPs compared to ConvMixer. The performance improvement of MDFF-ConvMixer is not just attributed to the increase in parameter count and FLOPs. MDFF-ConvMixer also achieves higher accuracy compared to a ConvMixer with an equivalent parameter count and FLOPs (ConvMixer + F) on various datasets. This observation is further substantiated in subsequent ablation experiments.

4.2.2 *Transformer models*

As seen from Table 1, on the multiscale small image datasets $\text{Dota}_{32\times32}$, $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$, $\text{cifar10}_{\frac{32}{2}\times\frac{32}{2}}$, and Fashion-MNIST, compared with the ViT that is not pretrained, Cross ViT and Deit, MDFF-ViT has obvious advantages in terms of its classification ability. Cross ViT can be regarded as an improved version of the ViT from the perspective of multiscale feature fusion, and compared with Cross ViT, MDFF-ViT yields accuracy improvements of 1.62%, 0.61%, 3.39%, and 2.23% on $\text{Dota}_{32\times32}$, $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$, $\text{cifar10}_{\frac{32}{2}\times\frac{32}{2}}$, and Fashion-MNIST, respectively, which shows that the improvement exhibited by MDFF-ViT over the ViT is not merely due to the fact that the number of network calculations increases. It also shows that MDFF is more effective than multiscale feature fusion under the same number of computations. On the four datasets, MDFF-ViT is stronger than the pretrained Cross ViT, and the classification ability of the pretrained ViT is competitive. The design of MDFF-ViT not only aims to improve accuracy but also considers the efficiency of the model. Due to the additional features provided by MDFF, MDFF-ViT requires fewer MLP layers and has a smaller parameter size compared to ViT and Cross ViT. Furthermore, the FLOPs of MDFF-ViT are only 1/7 of ViT's FLOPs, thanks to the convergence achieved by the multi-domain features of MDFF-ViT in a shallower network. In terms of accuracy, parameter size, and FLOPs, MDFF-ViT outperforms the non-pretrained ViT.

It is worth noting that MDFF-ViT, with its more complex feature fusion, demonstrates inferior recognition performance compared to the simpler feature fusion approach of MDFF-ConvMixer. This is because convolutional models, leveraging their inherent inductive prior for exploiting spatial invariance in 2D image data, outperform transformer-based models in recognition performance, with smaller parameter counts and computational requirements, especially in the case of small datasets and non-pretrained classification models. Consequently, when using non-pretrained models, ConvMixer outperforms ViT significantly in terms of recognition. As a result, MDFF-ConvMixer surpasses MDFF-ViT in recognition performance.

However, it should be acknowledged that MDFF achieves gains of 8.93%, 1.13%, 23.03%, and 3.90% for ViT on datasets $\text{Dota}_{32\times32}$, $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$, $\text{cifar10}_{\frac{32}{2}\times\frac{32}{2}}$, and Fashion-MNIST, respectively, whereas MDFF achieves gains of 2.32%, 1.29%, 3.86%, and 0.27% for ConvMixer on the same four datasets. Comparing the improvement rates of the enhanced algorithms with the baseline algorithms, it becomes evident that MDFF demonstrates significantly larger gains for ViT on datasets $\text{Dota}_{32\times32}$ and $\text{cifar10}_{\frac{32}{2}\times\frac{32}{2}}$, highlighting the effectiveness of more complex feature fusion.

4.3 Ablation Study

4.3.1 *Ablation Study with Each Improvements*

The MDFF-ConvMixer and MDFF-ViT frameworks designed in this paper both contain two improvements: the introduction of frequency domain features and their feature fusion modules. In this section, a series of ablation experiments are conducted with $\text{Dota}_{32\times32}$, $\text{Dota}_{\frac{32}{2}\times\frac{32}{2}}$, $\text{cifar10}_{\frac{32}{2}\times\frac{32}{2}}$, and Fashion-MNIST as experimental subjects to better understand the effectiveness

Table 2 Ablation study with each improvements on $Dota_{32\times32}$, $Dota_{\frac{32}{2}\times\frac{32}{2}}$ and $cifar10_{\frac{32}{2}\times\frac{32}{2}}$ and Fashion-MNIST.

Models	Params (M)	FLOPs (G)	$Dota_{32\times32}$ (%)	$Dota_{\frac{32}{2}\times\frac{32}{2}}$ (%)	$cifar10_{\frac{32}{2}\times\frac{32}{2}}$ (%)	Fashion MNIST (%)
			Convolutional			
ConvMixer	1.95	2.54	87.82	83.48	62.14	95.28
ConvMixer (D)	1.95	2.54	86.16↓1.66	80.61↓2.87	46.79↓15.35	86.83↓8.45
ConvMixer + F	3.90	5.09	88.50↑0.68	84.04↑0.56	64.26↑2.12	95.50↑0.22
ConvMixer + D + F (MDFF-ConvMixer)	3.90	5.09	**90.14↑2.32**	**84.77↑1.29**	**66.00↑3.86**	**95.55↑0.27**
			Transformer			
ViT	99.00	53.36	79.22	78.62	36.20	90.63
ViT(D)	99.00	53.36	77.60↓1.62	77.32↓1.30	33.65↓2.55	85.61↓5.02
ViT + F	38.62	7.96	86.53↑7.31	79.14↑0.52	55.84↑19.64	94.48↑3.85
ViT + D + F (MDFF-ViT)	38.62	7.96	**88.15↑8.93**	**79.75↑1.13**	**59.23↑23.03**	**94.53↑3.90**

Note: Bold values represent the best result.

of each improvement. In the experiment, A represents the algorithm name, D represents the frequency domain feature, and F represents feature fusion, which includes the following experimental combinations (Table 2).

1. A: no improvement points are added;
2. A (D): in the basic network structure, only frequency domain features are used, and spatial domain features are not used;
3. A + F: a dual-branch airspace feature fusion module is introduced to the basic network for investigating whether the achieved performance improvement only due to the increase in the number of parameters;
4. A + D + F: the MDFF-ConvMixer and MDFF-ViT methods proposed in this paper.

When D is added to the ViT and ConvMixer, since only the frequency domain features are used, the resulting effect is not as good as that yielded when only the spatial domain features are used, and the accuracy rate produced on the dataset decreases. When two improvement points are added, the classification accuracy improves the most, which demonstrates the effectiveness of the MDFF approach proposed in this paper.

4.3.2 *Ablation Study with Different Feature Fusion Methods of MDFF-ConvMixer*

Spatial domain features and frequency domain features are two different types of features, the better fusion of them, the more and richer information will be brought, which is very useful for classification. We have introduced some feasible feature fusion schemes based on ConvMixer respectively, details can be found in Sec. 3.3. In this section, to compare the performance of these fusion schemes and verify the effectiveness of our models, a series of experiments will be conducted on the $Dota_{32\times32}$ dataset. Except for these models, other hyperparameters such as batch size are the same as mentioned before. Same as below.

The results of different feature fusion schemes based on ConvMixer on $Dota_{32\times32}$ dataset are shown in Table 3, as well as their parameters and FLOPs. The symbols used in the table are related to Fig. 2. These schemes in the table correspond to our model MDFF-ConvMixer and five ConvMixer-related feature fusion schemes mentioned (from MC-Strategy1 to MC-Strategy5)

Table 3 Ablation study with different feature fusion schemes of ConvMixer on $\text{Dota}_{32\times32}$ dataset.

Models	Fusions	$\text{Dota}_{32\times32}$ (%)	Params (M)	FLOPs (G)
MDFF-ConvMixer	Concat y_{spa}^{M} and y_{fre}^{M}	**90.14**	3.90	5.09
MC-strategy1	Concat x_{spa} and x_{fre}	87.51	7.12	9.26
MC-strategy2	Concat y_{spa} and y_{fre}	88.63	7.65	9.94
MC-strategy3	Fuse y_{spa}^{M} and y_{fre}^{M} with attention	***89.86***	4.77	6.22
MC-strategy4	$\text{output}_{\text{spa}} + \text{output}_{\text{fre}}$	89.26	3.90	5.09
MC-strategy5	$\max(\text{output}_{\text{spa}}, \text{output}_{\text{fre}})$	87.38	3.90	5.09

Note: Bold value represents the best result, and bold italic value represents the suboptimal value.

in Sec. 3.3. As seen in the tabel, our MDFF-ConvMixer achieves the best accuracy with smallest FLOPs and parameters.

4.3.3 *Ablation Study with Different Patch Sizes Used in MDFF-ConvMixer*

When images are input into MDFF-ConvMixer, they are all processed into a sequence of embedded image-patches by patch embedding machine. Different patch sizes will have a great impact on model's performance. Here we will perform experiments on $\text{Dota}_{32\times32}$ dataset to understand the effect of patch sizes in MDFF-ConvMixer.

As a result of the feature fusion scheme of concate y_{spa}^{M} and y_{fre}^{M} is adopted in model MDFF-ConvMixer, the sizes of y_{spa}^{M} and y_{fre}^{M} must be the same, and enventually the patch sizes of the spa-branch and fre-branch must be the same, too. Since the patch sizes pair (7, 7) is used in MDFF-ConvMixer, we test the other four pairs of patch sizes on $\text{Dota}_{32\times32}$ dataset such as (3, 3); (5, 5); (9, 9); and (11, 11). Their accurateness, parameters, and FLOPs can be found in Table 4. The symbols used in the table are related to Fig. 2.The MC-Strategy6 to MC-Strategy9 means the four models mentioned above that contain different patch sizes. Smaller patch sizes lead to more FLOPs and richer information; bigger patch sizes reduce computation, but omit some details of targets, especially for small targets. Combined with the experimental results, by using the patch sizes pair (7, 7), MDFF-ConvMixer achieves the best accuracy with a little increase in parameters and FLOPs and it confirms the superiority of our model for small targets.

4.3.4 *Ablation Study with Different Feature Fusion Methods of MDFF-ViT*

Efficient feature fusion is the key to learn multi-domain feature representations. To confirm the effectiveness of MDFF-ViT, we propose other three different fusion strategies (from

Table 4 Ablation study with different path sizes of MDFF-ConvMixer on $\text{Dota}_{32\times32}$ dataset.

Models	Patch size		Depth		$\text{Dota}_{32\times32}$ (%)	Params (M)	FLOPs (G)
	spa	fre	spa	fre			
MDFF-ConvMixer	7	7	24	24	**90.14**	3.90	5.09
MC-strategy6	3	3	24	24	86.97	3.84	27.91
MC-strategy7	5	5	24	24	***89.30***	3.87	10.11
MC-strategy8	9	9	24	24	88.25	3.95	3.12
MC-strategy9	11	11	24	24	87.76	4.01	2.13

Note: Bold value represents the best result, and bold italic value represents the suboptimal value.

Table 5 Ablation study with different feature fusion schemes of ViT on $Dota_{32\times32}$ dataset.

Models	Fusions	$Dota_{32\times32}$ (%)	Params (M)	FLOPs (G)
MDFF-ViT	Cross-attention + score-add	**88.15**	38.62	7.96
MV-strategy1	Cross-attention + feature-concate	***87.38***	38.62	8.28
MV-strategy2	Score-add	86.88	26.02	6.66
MV-strategy3	Self-attention + score-add	87.11	44.92	9.93

Note: Bold value represents the best result, and bold italic value represents the suboptimal value.

MV-strategy1 to MV-strategy3), and test them on the $Dota_{32\times32}$ dataset, respectively. The details of each strategy are as follows.

MDFF-ViT: the method used in this article. First, fuse T_{spa} and T_{fre} by cross-attention module, and then send the two results $T^{cls}_{spa+fre}$ and $T^{cls}_{fre+spa}$ ouput by cross-attention module to two separate MLP heads to get two different classification scores, finally add the two classification scores to get the final result.

MV-strategy1: use cross-attention module to fuse T_{spa} and T_{fre}, then concat $T^{cls}_{spa+fre}$ and $T^{cls}_{fre+spa}$, which are outputs from cross-attention module, finally only use one MLP head to output the classification result;

MV-strategy2: compared with MDFF-ViT, only use transformer block for feature extraction, instead of cross-attention module for feature fusion;

MV-strategy3: compared with MDFF-ViT, only use self-attention module instead of cross-attention module to fuse features.

The symbols used above are related to Fig. 3. As seen in the Table 5, our MDFF-ViT achieves the best accuracy with minor increase in FLOPs and parameters.

4.3.5 *Ablation Study with Different Patch Sizes used in MDFF-ViT*

Different from MDFF-ConvMixer, MDFF-ViT adds the two classification scores of the double-branch as final output, so the sizes of T_{spa} and T_{fre} need not to be the same, which means we can set patch sizes differently for spa-branch and fre-branch in MDFF-ViT. We test 9 different patch

Table 6 Ablation study with different patch sizes of MDFF-ViT on $Dota_{32\times32}$ dataset.

Models	Patch size		Dimension		$Dota_{32\times32}$ (%)	Params (M)	FLOPs (G)
	spa	fre	spa	fre			
MDFF-ViT	16	16	384	384	**88.15**	38.62	7.96
MV-strategy4	32	32	384	384	87.36	40.24	2.21
MV-strategy5	32	16	384	384	87.39	39.44	5.25
MV-strategy6	32	8	384	384	87.21	39.51	17.54
MV-strategy7	16	32	384	384	87.03	39.44	5.25
MV-strategy8	16	8	384	384	87.56	38.70	20.58
MV-strategy9	8	32	384	384	87.10	39.51	17.54
MV-strategy10	8	16	384	384	***87.70***	38.70	20.58
MV-strategy11	8	8	384	384	86.70	38.77	32.87

Note: Bold value represents the best result, and bold italic value represents the suboptimal value.

Table 7 Comparisons with faster R-CNN on person-car dataset.

Models	Params (M)	FLOPs (G)	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_s (%)	AP_m (%)	AP_l (%)
Faster R-CNN	41.13	407.12	44.71	71.30	48.22	25.47	51.93	60.04
MDFF-faster R-CNN	55.91	465.00	**45.88**	**73.15**	**49.57**	**25.98**	**52.70**	**61.84**

Note: Bold values represent the best result.

sizes pairs on $Dota_{32\times32}$, respectively, to learn the effect of patch sizes, results are shown in Table 6. Without a doubt, MDFF-ViT achieves the best performance with the patch sizes pair of (16, 16). Intuitively, small patch sizes will increase model's computation and the memory usage of GPU, simultaneously big patch sizes will lose details. The patch sizes pair (8, 8) should get better results as it provides more fine-grained features; however it is not good as (16, 16) because of it's huge FLOPs. The patch size pair (8, 8) has 16 times as many tokens as patch sizes pair (16, 16). The large number of tokens will generate a lot of floating point operations (FLOPs) and take up a large amount of GPU memory, leading to a very small batch size, such as 1 and high randomness of the gradient of each layer of the model, which consumes a lot of training time and makes the model difficultly to converge.

5 Conclusion

We propose an MDFF method for small target classification, which realizes multidomain feature extraction through the fusion of frequency domain features and spatial domain features. The MDFF method enriches the information content of targets, which is crucial for improving the accuracy of small target classification tasks. Experiments demonstrate the effectiveness of this method. Although the current work in this study only involves research on small target classification, the MDFF idea presented in this work can be used in more computer vision fields theoretically, such as object detection. This is because in the task of object detection, networks often generate numerous proposals containing positive and negative samples. When performing bounding box regression on the proposals and ground truth bounding boxes (GT-bbox), it is necessary to

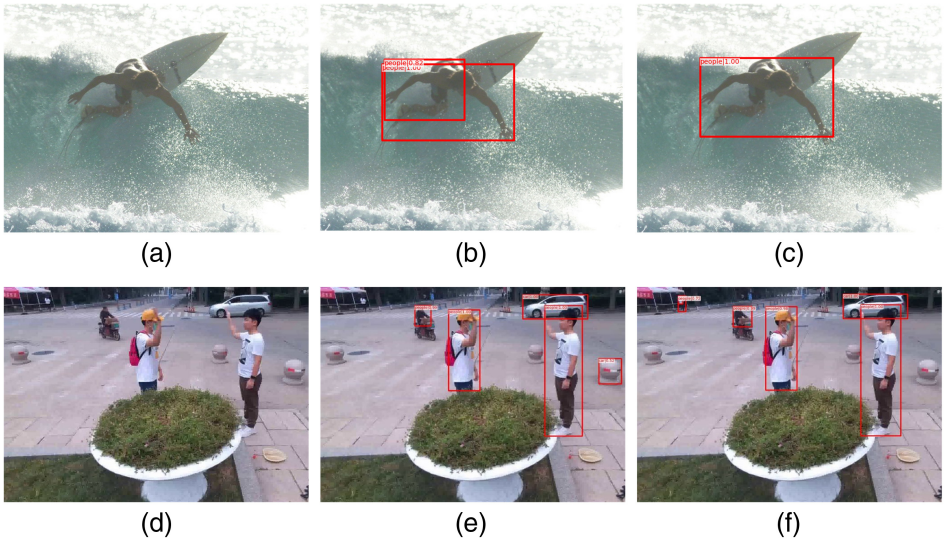

Fig. 6 Visualization of detection results. (a)–(c) The original image of person-car (a subset of COCO[34]), faster R-CNN[32] detection results, and MDFF-faster R-CNN[33] detection results, respectively. (d)–(f) The original image of UAV-human,[35] faster R-CNN detection results, and MDFF-faster R-CNN detection results, respectively. From panels (b) and (c), it can be observed that MDFF-faster R-CNN corrects the issue of duplicate detections present in faster R-CNN; from panels (e) and (f), it can be observed that MDFF-faster R-CNN corrects the false positive (misclassifying a stone as a car) and false negative errors of faster R-CNN.

assign a class to each positive proposal, a process similar to object classification. Therefore, the use of the MDFF method can increase the information contained in the proposals, which is highly beneficial for generating high-quality proposals. To demonstrate the feasibility of this viewpoint, we made a simple modification to the ROI-HEAD of faster R-CNN[32] by adding a frequency domain branch for class prediction and named it MDFF-faster R-CNN.[33] We trained it on a subset of the COCO[34] dataset (named person-car). To investigate the performance of our algorithm in real-world scenarios, we also conducted transfer learning on the unmanned aerial vehicle (UAV)-human[35] dataset for object detection, where the targets are observed from the perspective of UAVs. The experimental results in Table 7 confirmed that MDFF-faster R-CNN outperforms the original faster R-CNN in terms of detection performance. Despite the increase in network parameters, MDFF-faster R-CNN outperforms faster R-CNN in all COCO AP metrics, especially noteworthy is the improvement of 1.35% in AP_{75}, which further demonstrates that the MDFF method effectively enhances the quality of proposals. Furthermore, from the visualization results in Fig. 6, it can be observed that the inclusion of the MDFF method effectively eliminates false detections and improves the quality of predicted bounding boxes. Based on the experimental results, we are further convinced that incorporating multi-domain features will lead to better performance in object detection. We plan to conduct further research and investigation in subsequent studies to explore this extension thoroughly.

Code, Data, and Materials Availability

The truth dataset used in our study is publicly available. The author's code and dataset are not publicly available at this time but are available from the authors upon reasonable request.

Acknowledgments

This work has been partly supported by funding received from the National Key Laboratory of Science and Technology on Multi-Spectral Information Processing through the Research on Bee Colony Target Behavior Analysis Technology program (Grant No. 6142113210408).

References

1. R. Moradi, R. Berangi, and B. Minaei, "Sparsemaps: convolutional networks with sparse feature maps for tiny image classification," *Expert Syst. Appl.* **119**, 142–154 (2019).
2. S. Hu et al., "ACCV: automatic classification algorithm of cataract video based on deep learning," *Biomed. Eng. Online* **20**(1), 1–17 (2021).
3. Y. Yang et al., "Perceptual face inpainting with multicolumn gated convolutional network," *J. Electron. Imaging* **31**(1), 013022 (2022).
4. Z.-Q. Li et al., "Learning efficient structured dictionary for image classification," *J. Electron. Imaging* **29**(3), 033019 (2020).
5. H.-F. Yin and X.-J. Wu, "Class-specific residual constraint non-negative representation for pattern classification," *J. Electron. Imaging* **29**(2), 023014 (2020).
6. K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 770–778 (2016).
7. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, pp. 1–14 (2014).
8. G. Huang et al., "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pp. 4700–4708 (2017).
9. K. Xu et al., "Learning in the frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 1740–1749 (2020).
10. A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst. 30* (2017).
11. J. Devlin et al., "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Association for Computational Linguistics, Minneapolis, Minnesota (2019).
12. N. Parmar et al., "Image transformer," in *Int. Conf. Mach. Learn.*, PMLR, pp. 4055–4064 (2018).
13. A. Dosovitskiy et al., "An image is worth 16x16 words: transformers for image recognition at scale," in *Int. Conf. Learn. Represent.* (2020).

14. C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vision*, pp. 357–366 (2021).
15. A. Trockman and J. Z. Kolter, "Patches are all you need?," https://openreview.net/forum?id=TVHS5Y4dNvM (2022).
16. Y. Hsu et al., "Pattern recognition experiments in the mandala/cosine domain," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-5**(5), 512–520 (1983).
17. B. Shen and I. K. Sethi, "Direct feature extraction from compressed images," *Proc. SPIE* **2670**, 404–414 (1996).
18. M. Ehrlich and L. S. Davis, "Deep residual learning in the JPEG transform domain," in *Proc. IEEE/CVF Int. Conf. Comput. Vision*, pp. 3484–3493 (2019).
19. L. Gueguen et al., "Faster neural networks straight from jpeg," in *Adv. Neural Inf. Process. Syst. 31* (2018).
20. S. Arber et al., "MLP-deficient mice exhibit a disruption of cardiac cytoarchitectural organization, dilated cardiomyopathy, and heart failure," *Cell* **88**(3), 393–403 (1997).
21. M. Xu et al., "Learning EEG topographical representation for classification via convolutional neural network," *Pattern Recognit.* **105**, 107390 (2020).
22. Z. Huang et al., "CCNet: criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vision*, pp. 603–612 (2019).
23. S. Misra and R. H. Laskar, "Approach toward extraction of sparse texture features and their application in robust two-level bare-hand detection," *J. Electron. Imaging* **27**(5), 051209 (2018).
24. M. M. Asem, I. S. Oveisi, and M. Janbozorgi, "Blood vessel segmentation in modern wide-field retinal images in the presence of additive gaussian noise," *J. Med. Imaging* **5**(3), 031405 (2018).
25. A. Saleem et al., "Segmentation and classification of consumer-grade and dermoscopic skin cancer images using hybrid textural analysis," *J. Med. Imaging* **6**(3), 034501 (2019).
26. A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's Thesis, Univ. of Tront (2009).
27. H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," arXiv:1708.07747 (2017).
28. Y. Rao et al., "Hornet: efficient high-order spatial interactions with recursive gated convolutions," in *Adv. Neural Inf. Process. Syst.*, Vol. 35, pp. 10353–10366 (2022).
29. H. Touvron et al., "Training data-efficient image transformers & distillation through attention," in *Int. Conf. Mach. Learn.*, PMLR, pp. 10347–10357 (2021).
30. Z. Liu et al., "Swin transformer: hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vision*, pp. 10012–10022 (2021).
31. X. Dong et al., "Cswin transformer: a general vision transformer backbone with cross-shaped windows," in *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 12124–12134 (2022).
32. S. Ren et al., "Faster R-CNN: towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst. 28* (2015).
33. Z. Zhang, "Research on small object recognition and detection technology based on deep learning," Master's thesis, Huazhong University of Science and Technology (2023).
34. T.-Y. Lin et al., "Microsoft coco: common objects in context," *Lect. Notes Comput. Sci.* **8693**, 740–755 (2014).
35. T. Li et al., "UAV-human: a large benchmark for human behavior understanding with unmanned aerial vehicles," in *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 16266–16275 (2021).

Jing Hu received her BSc degree in communication engineering and her MSc and PhD degrees in control engineering from Huazhong University of Science and Technology, Wuhan, China, in 1999, 2002, and 2010, respectively. Currently, she is an associate professor in multi-spectral information processing with the National Key Laboratory of Science and Technology, Huazhong University of Science and Technology. Her research interests include statistical pattern recognition and image analysis.

Zican Shi received his bachelor's degree in engineering from the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China, in 2022. Currently, he is pursuing his master's degree at the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China. His research interests include small object recognition and small object detection.

Zheng Zhang received his bachelor's degree in communication engineering from Jilin University, Changchun, China, in 2020. Currently, he is pursuing his master's degree at the School of Artificial Intelligence and Automation, Huazhong University of Science and

Technology, Wuhan, China. And he will get his master's degree in July 2023. His research interests include small object recognition and small object detection.

Siqi Lv received her bachelor's degree in science from the School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan, China, in 2021. Currently, she is pursuing a master's degree at the School of Artificial Intelligence and Automation Huazhong University of Science and Technology, Wuhan, China. Her research interests include small target classification and small target detection.

Yifan Chen received his bachelor's degree in engineering from the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China, in 2022. Currently, he is pursuing his master's degree at the School of Artificial Intelligence and Automation Huazhong University of Science and Technology, Wuhan, China. His research interest includes group target tracking.

Yan Ouyang received his BE and ME degrees of software engineering and computer science from Wuhan University of Technology, China, in 2006 and 2009, respectively. He received his PhD of control science and engineering from Huazhong University of Science and Technology, Wuhan, China, in 2013. His research interests include facial expression analysis, computer vision, and pattern recognition. In recent years, he is mainly working on the application of sparse representation theory in facial expression recognition.

Jia He holds a master's degree. Currently, he is an engineer of the 95841 army, mainly engaged in the testing and verification of the infrared characteristics of the target and the environment.