Proceedings of the 3rd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2025.22704

Few-shot Image Information Mining and Its Application in UAV Early Warning

Miaorui Wang

Harbin Institute of Information Technology, Harbin, China
15031758978@163.com

Abstract: Artificial intelligence applications in vertical fields face the problem of a small and unbalanced number of annotated samples. This article attempts to solve the small sample problem from three perspectives: data, model, and algorithm, by selecting UAV early warning. For data, conventional image enhancement, Mixup enhancement, random masking, and Partial convolution techniques are used to enhance the paradigm of Few-shot samples. Transfer learning techniques are applied to algorithms. Firstly, an autoencoder is trained using many unlabeled images combined with a masking strategy. Secondly, a classification layer is added at the end of the encoding part of the autoencoder. For the model, a multitasking strategy is adopted, which involves both supervised classification and self-supervised image inpainting of annotated images to improve the accuracy of the supervised classification part of the network after transfer learning. Finally, this article draws on the k-nearest neighbor algorithm and metric-based meta-learning ideas and introduces an anomaly warning mechanism, which is combined with conventional detection and warning to reduce the missed alarm rate in UAV early warning.

Keywords: Few-shot Learning, Autoencoder, Partial Convolution, Mixup Data Augmentation, UAV Early Warning

1. Introduction

Deep learning technology has developed rapidly over the past decade [1]. For example, in 2011, the activation function ReLU was proposed, replacing the original sigmoid activation function; in 2012, the AlexNet network, which triggered the CV AI wave, was proposed, and in 2013, word2vec was proposed in the NLP field; in 2014, there were more achievements in deep learning, with the invention of Generative Adversarial Networks (GAN) and the Adam optimizer, and in the NLP field, from the invention of Transformer [2] in 2017 to the release of ChatGPT by OpenAI in 2022, the development of large models has since profoundly influenced various aspects of artificial intelligence. With the continuous progress of deep learning technology, artificial intelligence has achieved remarkable success in vision and image [3], natural language [4], speech [5], and multi-modal fields [6]. The achievements of artificial intelligence are inseparable from the support of a vast amount of valuable data, both labeled and unlabeled. How to better utilize deep learning technology without sufficient data has become a hot issue accompanying the development of deep learning, commonly known as the few-shot and low-shot problems.

Notwithstanding these successes, many specialized domains—ranging from construction and finance to energy—lack extensive, high-quality datasets of the sort found in widely studied areas

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.22704

like internet-scale data or large medical repositories. Consequently, effective deployment of AI in these sectors is hampered by “few-shot” or “low-shot” problems, where training data are sparse or imbalanced [7]. Researchers have approached this challenge through three interconnected perspectives: data-centric, model-centric, and algorithm-centric solutions.

Data augmentation is the most direct method to increase sample data volume in deep learning. Using prior knowledge to increase sample quantity means expanding the paradigm of deep learning input data, reducing the model's estimation error. Data augmentation can be divided into single-sample data augmentation and multi-sample data augmentation, such as flipping, rotating, cropping, scaling, and masking an image, which belong to single-sample data augmentation, while multi-sample data augmentation uses multiple samples to generate new samples, such as Smote, SamplePairing, and Mixup techniques. Deep learning technologies such as unsupervised learning, semi-supervised learning, active learning, generative artificial intelligence, and large models also provide more possibilities for data augmentation. Additionally, data augmentation is not an independent few-shot strategy and needs to be combined with deep learning technologies, such as the pseudo-label learning method in semi-supervised learning [8], which can also be considered an example of data augmentation.

In this work, we choose UAV early warning as a representative application to demonstrate how data, model, and algorithm strategies may be integrated to address few-shot challenges. We employ conventional image augmentation, random masking, and partial convolutions on the data side, a multi-task approach for the network architecture, and transfer learning coupled with metric-learning insights to mitigate potential missed detections. In particular, k-nearest neighbor strategies and meta-learning–inspired techniques serve to lower false-negative rates, underscoring the importance of holistic solutions for few-shot scenarios in real-world AI applications.

2. Literature Review

2.1. Pre-trained Models

In 1986, Rumelhart, Hinton, and Williams first proposed the autoencoder [9]. Autoencoder networks have a wide range of applications in artificial intelligence [10], such as image coloring, image denoising, image compression, and large model training. Autoencoders are divided into convolutional autoencoders, sparse autoencoders, and contractive autoencoders, with convolutional autoencoders being the most used because they are more suitable for image processing and feature extraction. This article adopts a convolutional autoencoder as the pre-trained model, with a structure consisting of an encoder, decoder, and bottleneck (see Figure 1), and uses a self-supervised training method to extract the most representative latent features from the input image, i.e., the features contained in the bottleneck part of Figure 1.

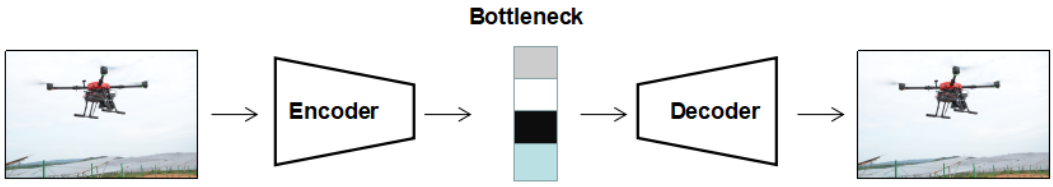


Figure 1: Autoencoder

The specific components of Figure 1 are shown in Figure 2. The overall architecture is a U-Net-like network structure, but since this article uses an autoencoder for purposes other than reconstructing the input image, the skip connection layer is omitted. In the encoder part, down sampling is not achieved through pooling layers but through the stride of convolution operations.

DOI: 10.54254/2755-2721/2025.22704

Additionally, the convolution layers use partial convolution [11], mainly to cooperate with the masking strategy in image augmentation. In the decoder part, the convolution layers are still conventional convolutions, and feature map upsampling uses common upsampling layers.

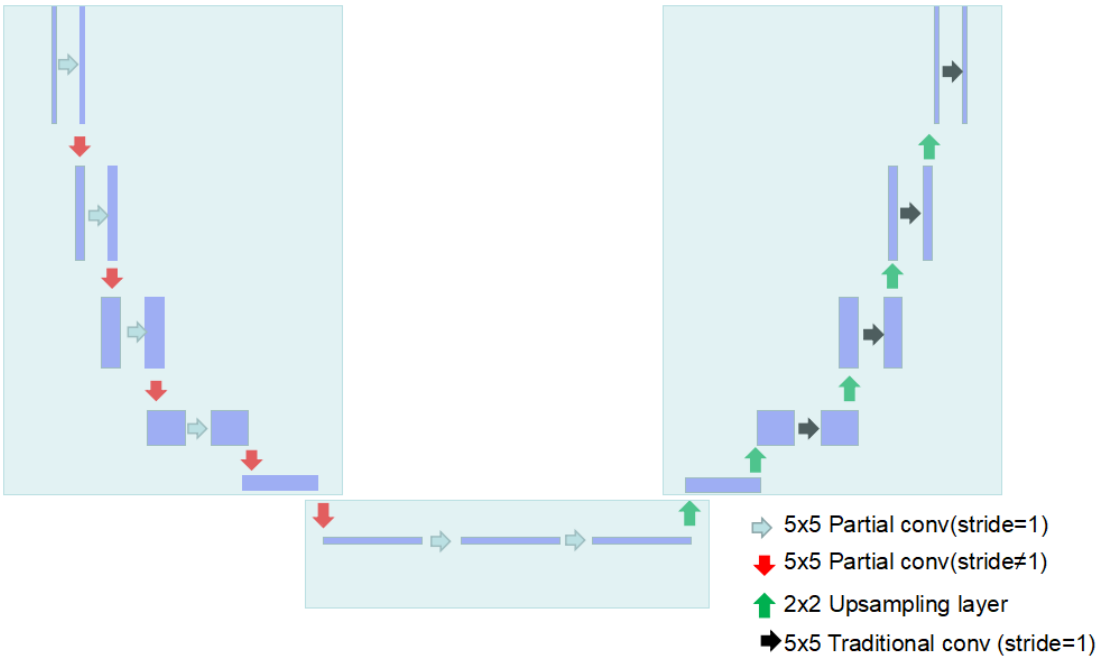


Figure 2: Structure of the Autoencoder Network

2.2. Transfer Networks

Based on the pre-trained autoencoder network, a multi-task transfer network can simultaneously perform classification and image inpainting tasks (see Figure 3). To achieve image classification, the network in Figure 3 adds a classifier composed of convolutional and fully connected layers to the bottleneck part of Figure 1 or Figure 2. Specifically, during transfer training, the parameters of the encoder, bottleneck, and decoder inherit those of the pre-trained model, but the parameters of the first half of the encoder are set to non-trainable.

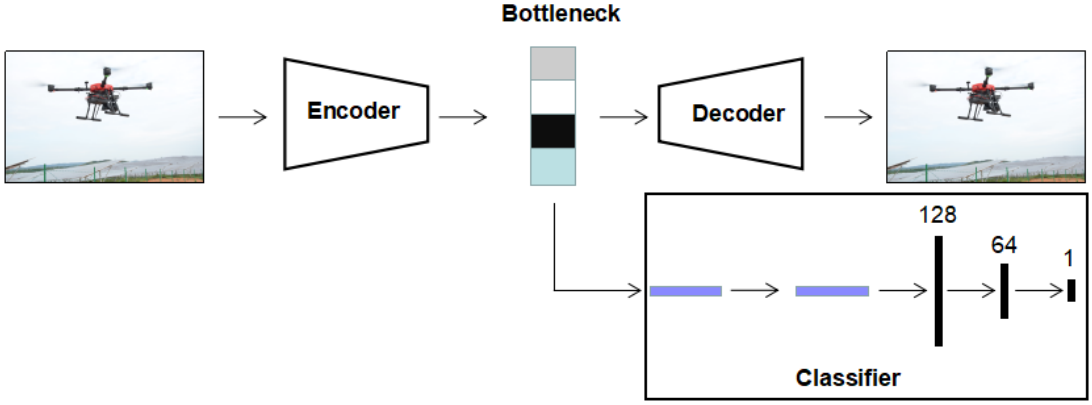


Figure 3: Autoencoder with Classifier

The design of the multi-task network in Figure 3 retains image inpainting for two purposes: first, to enhance the encoder's ability to extract image features through the image inpainting task, making

DOI: 10.54254/2755-2721/2025.22704

the extracted latent features better serve the classifier; and second, to cooperate with the image masking augmentation strategy, implementing the image masking strategy through image inpainting. In summary, the two non-exclusive tasks are carried out simultaneously within the same network, which reduces the model's complexity and allows the classifier to perform better.

The objective function of the multi-task network consists of two parts: one is image inpainting using mean-squared error, and the other is the classifier using binary cross-entropy, with a weight ratio of 1:1.

2.3. Partial Convolution

To increase the diversity of few-shot images, the autoencoder part in the figure uses partial convolution. Randomly masking and inpainting images is a common self-supervised model training strategy in the field of artificial intelligence images [12]. The random masking strategy not only increases image diversity in VIT but also in convolutional models and can be considered a data augmentation method. Xu Haohang further developed a data augmentation method based on this [13]. Suppose conventional convolution is used for image inpainting. In that case, the convolution operation indiscriminately calculates the damaged and undamaged regions of the image, which has two problems: first, the damaged region (assuming it is filled with 0) causes most of the calculations in the damaged region to be wasted, and second, the damaged region filled with zero values is also input into the convolution layer, making the convolution less sensitive to the information difference between the two parts. Therefore, Liu G developed a partial convolution technique in literature [11] to deal with image damage, which directly introduces a masking matrix to define the damaged part of the image as invalid pixels. Liu G further applied partial convolution to the common operation of padding in convolution in literature [14].

The calculation process of partial convolution is as follows:

$$x' = \begin{cases} W^T(X \otimes M)\dfrac{sum(1)}{sum(M)} + b & \text{if } sum(M) > 0 \\ 0 & \text{otherwise} \end{cases}$$

X is the input feature, M is the input masking matrix, W is the convolution kernel, b is the bias, and 1 is the identity matrix.

The update rule for the masking matrix M is as follows:

$$m' = \begin{cases} 1 & \text{if } sum(M) > 0 \\ 0 & \text{otherwise} \end{cases}$$

That is, if one element of M is 1, then $m'$is 1.

Figure 4 provides a detailed calculation process of partial convolution. In Figure 4, the top row of images represents the assumed input data, masking matrix, convolution kernel, identity matrix, and bias. Under the condition of Padding being Same, the input data needs to be padded, and the masking matrix is also padded in the same way, i.e., with 0s. By strictly following the above partial convolution calculation process and masking matrix update rules, the result of partial convolution and the updated masking matrix can be obtained. Figure 4 only illustrates the calculation process of three elements in the top left, middle, and bottom right, represented in different colors.

DOI: 10.54254/2755-2721/2025.22704

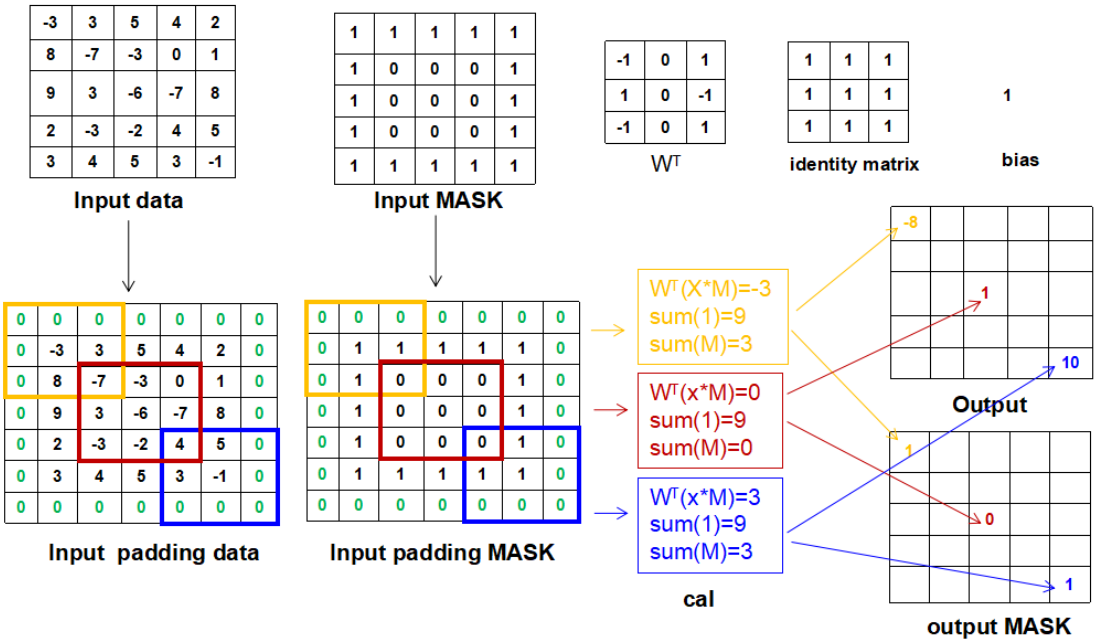


Figure 4: Calculation Process of Partial Convolution

3. Methods

3.1. Data Collection

This article selects the CIFAR100 image library to perform self-supervised pre-training on the network shown in Figure 2. It then selects the Intel Natural Scenery Image Classification dataset(http://suo.nz/2KwdmY) and 418 UAV images to perform transfer learning based on the above pre-trained parameters. The Intel Natural Scenery Image dataset consists of 24,335 images, sized at 150x150, which, compared to UAV images, belong to a large sample dataset and are closer to the background of UAV images. This is the reason for choosing the Intel Natural Scenery Image dataset instead of other images. To test the effectiveness of the algorithm, 300 other natural scenery images and 120 UAV images were also selected from the web as test data. In this article, images containing UAVs are referred to as few-shot images, while the rest, such as scenery images, are referred to as large-sample images.

3.2. Data Augmentation

Image augmentation is first considered for the 418 UAV images, which are considered few-shot. This article selects three enhancement methods: single-sample conventional augmentation, multi-sample Mixup augmentation, and image masking. Conventional augmentation involves flipping the images horizontally and adding fog, clouds, coarse pepper noise, salt-and-pepper noise, random cropping, random channel loss, and rotation to the original images and their horizontally flipped versions. After conventional augmentation, the images are expanded to 16 times the original, or 5,852 images. The specific results of conventional augmentation are shown in Figure 5.

DOI: 10.54254/2755-2721/2025.22704

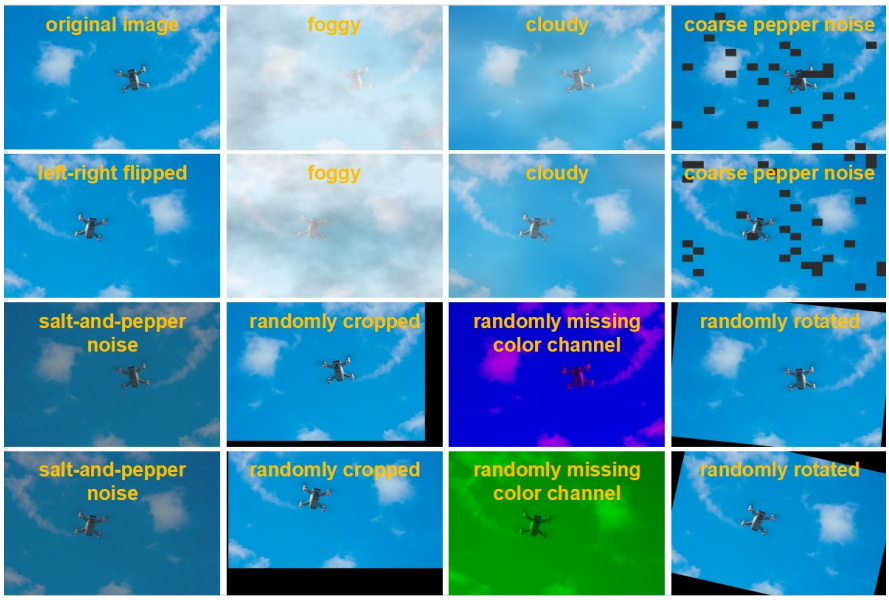


Figure 5: Results of Single-Sample Image Augmentation

Mixup technology generates new samples by adding two samples in proportion. This article uses two methods to generate new UAV samples. The first method selects all UAV images to synthesize new samples, with the labels still being UAV images. The second method selects UAV images and Intel Natural Scenery images, with the synthesis weight for UAV images being greater than 0.5, so the labels are still UAV images. The specific formulas are as follows:

$$\tilde{x} = \lambda x_i + (1 - \lambda)\lambda x_j$$

$$\tilde{y} = y_i$$

where $\lambda \in (0.5, 1)$ is the probability value, (x_i, y_i) is the UAV sample, and (x_j, y_j) is a different UAV sample or an Intel Natural Scenery image sample. By performing Mixup augmentation on the 418 UAV images and their horizontally flipped images four times, 3,344 new labeled data are obtained. Figure 6 shows the results of Mixup image augmentation, with the top three images in Figure 6 demonstrating new UAV images generated using the first method, and the bottom three images showing new UAV images generated using the second method.

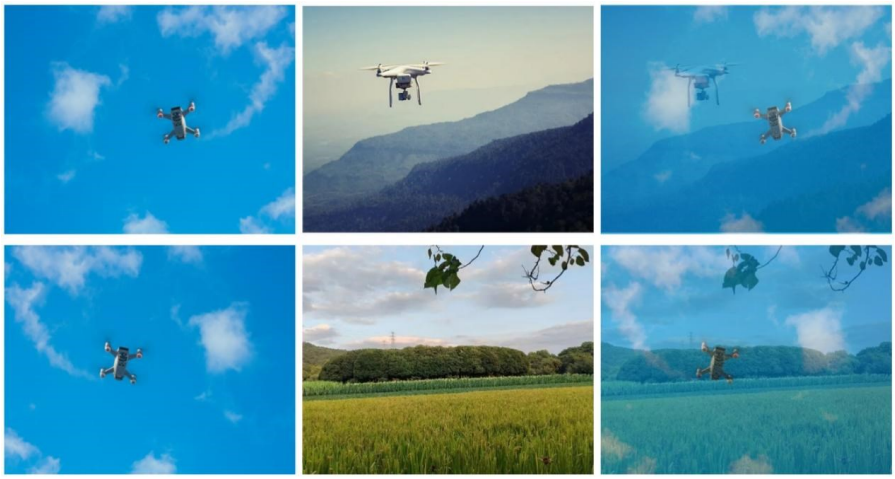


Figure 6: Results of Multi-Sample Mixup Image Augmentation

DOI: 10.54254/2755-2721/2025.22704

Through single-sample conventional augmentation and multi-sample Mixup augmentation, the original 418 images are increased to 24 times the original, that is 10,032 images.

To enhance data diversity, image masking is applied using four methods: rectangular, straight line, solid circle, and annulus. The masking is randomly changed during training and is applied to both large-sample and few-shot images. The image masking effect is shown in Figure 7.

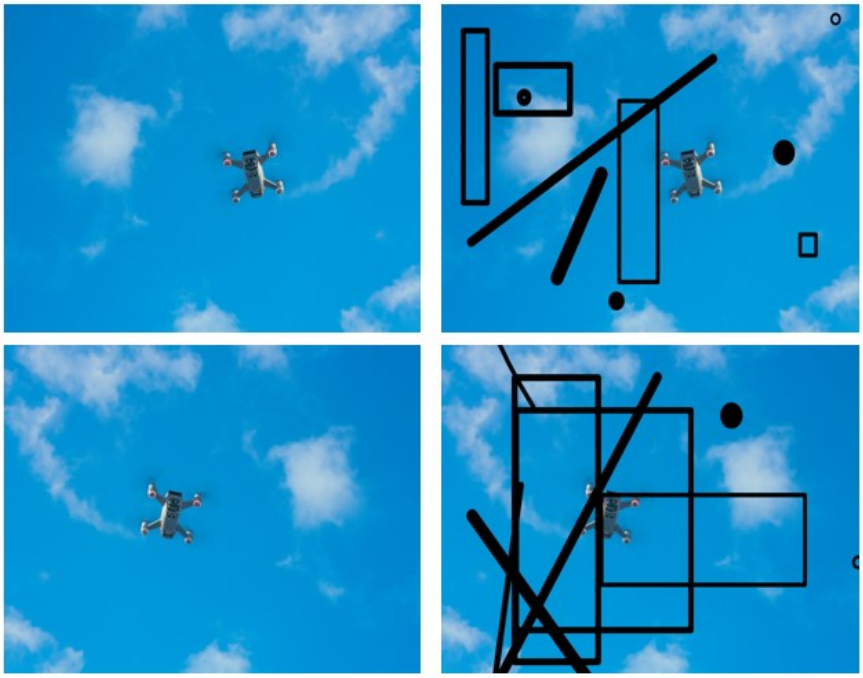


Figure 7: Results of Image Masking

4. Results

4.1. Comparative Experiments

This study organizes three comparative experiments focusing on 1) comparing traditional convolution and partial convolution, 2) comparing single-task and multi-task learning, and 3) examining the effect of transfer learning.

In the first experiment, the training phase using partial convolution combined with masking broadens data diversity. As shown in Tables 1 and 2, this leads to slightly better inference performance than traditional convolution. Specifically, on a test set of 300 natural scenery images and 120 UAV images, raditional convolution correctly identifies 283 and 113 images, respectively, whereas partial convolution correctly identifies 293 and 116 images. Table 2 converts the results from Table 1 into accuracy, precision, recall, and F1 scores.

In the second experiment, both the single-task and multi-task models undergo the same self-supervised pre-training on the CIFAR100 image set. When performing transfer learning on the Intel Natural Scenery and UAV images, the objective functions used are different. Single-task training only uses supervised classification with cross-entropy as the objective function, while multi-task training uses both supervised classification and self-supervised image inpainting, with the objective function being the sum of cross-entropy and mean-squared error, in a ratio of 1:1. Through inference on 420 images, the network trained with multi-task training achieved better results (see Tables 1 and 2). The main reason is that the image inpainting task improved the accuracy of the network's encoding part from another dimension, compressing the model search space for the image

DOI: 10.54254/2755-2721/2025.22704

classification task, thus verifying the effectiveness of using a multi-task strategy to solve the few-shot problem from the model perspective.

In the third experiment, evaluating transfer learning involves initially pre-training the network (see Figure 2) on the CIFAR100 dataset using image inpainting, then importing these pre-trained parameters into the training network (see Figure 3). Specifically, the parameters of the first half of the decoder are frozen, while the parameters of the second half of the decoder, the bottleneck, the added classification task network, and the encoder are set to trainable. Without using a pre-trained model, training is performed directly on the network in Figure 2. Through inference on 420 images, the results showed a slight improvement in accuracy when using transfer technology (see Tables 1 and 2). Although there was only a one-image improvement in the recognition accuracy of 300 natural scenery images and a two-image improvement in the recognition accuracy of 120 UAV images when using transfer technology compared to not using it, the training process using a pre-trained network was much more efficient than that using a network with completely random parameters. Moreover, with the development of large models and other technologies, it has become a common practice to use unsupervised, self-supervised, or semi-supervised methods to pre-train models and then fine-tune them for downstream tasks, and their effectiveness is self-evident.

Table 1: Results of 420 Images by Different Methods

Method\Test images	**300 Natural images**	**120 UAV images**
Traditional Convolution	283	113
Single-task	281	111
Without Transfer	292	114
Ours	293	116

Table 2: Evaluation Criteria of 420 Images by Different Methods

Method\Criteria	Accuracy	Precision	Recall	F1 Score
Traditional Convolution	94.29	86.92	94.17	90.40
Single-task	93.33	85.38	92.50	88.80
Without Transfer	96.67	93.44	95.00	94.21
Ours	97.38	94.31	96.67	95.47

The comparative experiments essentially compare three aspects. The comparison between traditional convolution and partial convolution is essentially a comparison of whether image masking is used, which is a comparison of data augmentation. The comparison between single-task and multi-task is a comparison from the model perspective, while the comparison of whether transfer technology is used is a comparison from the algorithm perspective. Although the three approaches to solving the few-shot problem are different and the effects vary, their comprehensive application is an effective strategy for solving the few-shot problem.

4.2. Visualization Analysis

This article uses the evolved version of CAM, Grad-CAM [16-17], and latent feature analysis to visualize the network used in this article. First, the sensitive regions of the original image in the network's convolutional layers when the output class is a UAV are analyzed to prove that the network has extracted key UAV information. The results are shown in Figure 8.

DOI: 10.54254/2755-2721/2025.22704

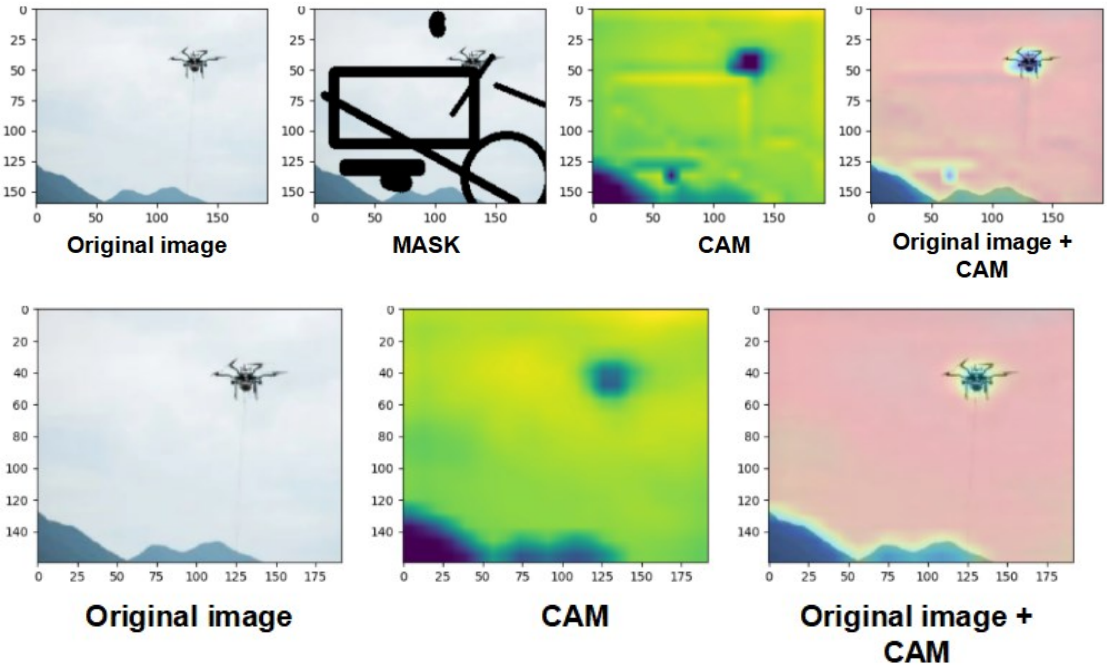


Figure 8: Visualization Analysis of the Network

Although the masking strategy increases data diversity during training, which is conducive to training a more robust network, Figure 8 shows that masking has some impact on the network's extraction of key image information, which is also the reason for setting the mask to all 1s during inference.

A layer was added to the classification part of this article's network to extract a 128-length feature vector, which can be considered as the encoded features of the image. The 128-length distributions corresponding to landscape images and UAV images would be different. To more intuitively demonstrate that the classifier in this article's network has learned the image classification information, the 128-length features were reduced to 2-length features using PCA and plotted on a single graph. The results are as follows.

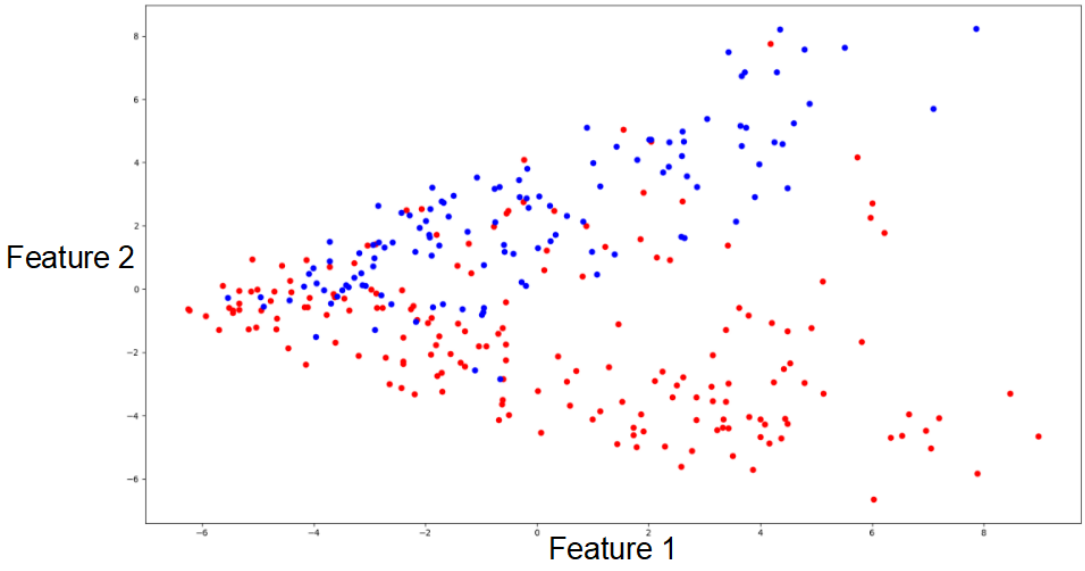


Figure 9: PCA Results for 128-Length Feature Vectors

DOI: 10.54254/2755-2721/2025.22704

In Figure 9, red represents UAV images, and blue represents landscape images. The feature vectors of UAV images and landscape images are distinguishable in space, although the effective information loss during the PCA process from 128 vectors to 2 vectors prevents them from being completely separated.

4.3. Reducing the UAV False Negative Alarm Rate

In UAV early warning AI scenarios, the risk of missed detections is far more critical than false alarms. It is better to sacrifice some alarm accuracy than to miss a UAV alarm. For this reason, especially under few-shot conditions, it is not possible to rely solely on the deterministic classification detection and warning mentioned in the previous chapters. An anomaly-based warning strategy must also be used to reduce the false negative alarm rate of UAVs.

The first anomaly-based warning strategy is called the k-nearest neighbor warning strategy. It determines the class affiliation of a test sample by calculating the distance between it and all other samples in the sample library. The distance used in this article is cosine similarity. Specifically, the 128-length latent feature vector obtained by an input image through the bottleneck part of the model in Figure 3 is calculated, and its cosine similarity with all images in the known annotated library is calculated. The average cosine similarity distance is used to determine whether an alarm is needed. For example, the blue triangle in the left figure of Figure 10 is like a small part of the large-sample images in the library. The classification network also determines that it belongs to the large-sample images, but its average similarity distance to the few-shot images is closer, so it should be determined to belong to the few-shot image class and trigger an alarm. The second is called the large-sample anomaly alarm strategy, which only calculates the class center of the feature length of the large-sample images. When a new sample is far from this class center, we determine that the sample is likely a few-shot image. This is similar to the ProtoNet method in meta-learning. In actual scenarios similar to this, it is like using artificial intelligence for product defect detection. Since defective products are a minority and qualified products are the majority, deep learning can learn the distribution characteristics of qualified products. If a product's feature distribution is different from that of qualified products, it can be determined to be a defective product. The right figure of Figure 10 assumes that there are two feature vectors. The blue triangle is located at the edge of the feature distribution of the large-sample images. A certain threshold can be set to determine that it is a few-shot image. This approach has the drawback of increasing the risk of misclassifying what should be large-sample images as few-shot images (red points).

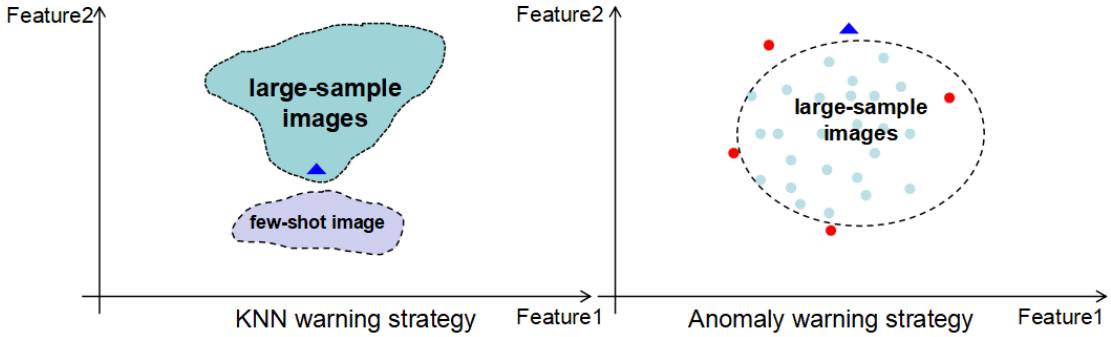


Figure 10: Two Methods of UAV Early Warning

The above two methods for reducing the false negative alarm rate of UAVs are somewhat of a last resort. After these two warning strategies, 281 out of 300 landscape images were predicted correctly, and 119 out of 120 UAV images were predicted correctly. The recall metric has

DOI: 10.54254/2755-2721/2025.22704

significantly increased, but the precision metric has dropped significantly. That is, accuracy has been sacrificed to ensure the false negative alarm rate of UAVs.

5. Conclusion

This work addresses the challenge of few and imbalanced annotated samples in deep learning by integrating solutions at the data, model, and algorithmic levels. On the data side, we adopt standard and Mixup-based augmentations, supplemented by random masking to enrich the training set. From a modeling perspective, we develop a combined autoencoder framework for image classification and inpainting, thereby enhancing feature extraction and classification accuracy. Furthermore, transfer learning is applied using a pre-trained autoencoder, accelerating training and improving performance when dealing with limited data. Partial convolution complements the masking strategy, boosting robustness in the face of occlusions or missing image regions.

To improve UAV early warning accuracy in real-world conditions, we merge a conventional classification approach with a feature-based anomaly detection mechanism, thereby reducing the likelihood of missed alarms. While the study demonstrates tangible benefits for mitigating few-shot and class imbalance issues, challenges remain—chief among them, the relatively high computational cost of the k-nearest neighbor strategy and the precision trade-offs introduced by anomaly-based detection. Advancing these methods will necessitate broader use of unlabeled data, semi-supervised or unsupervised techniques, and large-model innovations, ultimately paving the way for more efficient and robust AI deployments in specialized domains.

References

[1] *Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural Computati on, 18(7), 1527–1552.*

[2] *Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 5998–6008.*

[3] *He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[4] *Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv. https://doi.org/10.48550/arXiv.1810.04805.*

[5] *Can, D., & Saraçlar, M. (2011). Lattice indexing for spoken term detection. IEEE Transactions on Audio, Speech, and Language Processing, 19(8), 2338–2346.*

[6] *Gangi Reddy, R., Rui, X., Li, M., & Bansal, M. (2021). MuMuQA: Multimedia multi-hop news question answering via cross-media knowledge extraction and grounding. arXiv. https://doi.org/10.48550/arXiv.2112.10728.*

[7] *Bendre, N., Marín, H. T., & Najafirad, P. (2020). Learning from few samples: A survey.*

[8] *Lee, D. H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. ICML 2013 Workshop on Challenges in Representation Learning.*

[9] *Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536.*

[10] *Michelucci, U. (2022). An introduction to autoencoders. Springer.*

[11] *Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., & Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. Proceedings of the European Conference on Computer Vision.*

[12] *He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). Masked autoencoders are scalable vision learners. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[13] *Xu, H., Ding, S., Zhang, Y., & Liu, X. (2022). Masked autoencoders are robust data augmentors..*

[14] *Liu, G., Shih, K. J., Wang, T.-C., & Reda, F. A. (2018). Partial convolution based padding.*

[15] *Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

[16] *Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International Conference on Computer Vision.*