Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ31147

Investigating Foreground Background Separation in Vision Transformers for Image Classification

Bowen Jiang

McCormick School of Engineering, Northwestern University, Evanston, USA
1179685746@qq.com

Abstract. Image categorization is a key part of computer vision. It may be used for a wide range of things, from self-driving cars to medical imaging. Most traditional methods look at an image as a whole, which makes it hard for them to pick up on the different contributions of front items and background context. Contour information, which is very important for finding important structures, is often not used enough. Vision Transformers (ViTs) and other recent developments have greatly improved classification performance, but they still depend on a single representation of the image. In this study, we investigate the potential advantages of deliberately segregating picture components for categorization purposes. We suggest a dual-stream ViT framework that works with foreground and background areas separately before putting their representations together. The experimental results indicate that the suggested dual-stream model does not surpass the performance of ordinary single-stream ViTs, but rather demonstrates equivalent efficacy across several benchmarks. More examination shows that the fundamental problem is that it is hard to separate the foreground from the backdrop. In complicated or messy scenarios, improper region extraction adds noise that makes the dual-stream approach less useful. These results show that component-aware designs have a lot of potential, but their success depends a lot on how well foreground–background segmentation works, which is still a big problem for the future.

Keywords: Image classification, Transformer, ImageNet, Deep learning

1. Introduction

Humans can quite easily distinguish the outlines of objects, but for computers, it's a really tough task. In image recognition, contour info is super important for spotting and telling apart different objects and scenes. David Marr talked about this in his book "Vision". He came up with a framework for visual processing which includes handling contour info - it's a key part of his theory [1]. Even though there have been some good algorithms like SAM (Structure-aware Matching) that have made progress in contour recognition recently, they still can't fully figure out the contours of every single object.

Algorithms can distinguish small elements of an item, such a dog's ears, legs, or body, in many computer vision applications. But it is still hard to tell the object apart from the complicated environment around it. This restriction underscores a major challenge in modern vision systems: the recognition of objects in disordered or chaotic environments. Image categorization is a traditional

© 2026 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: 10.54254/2755-2721/2026.TJ31147

challenge in computer vision, typically addressed by partitioning an image into foreground and background segments [2]. In most situations, study focuses on the foreground since it has the most important semantic information about the image, while the background is there to give context.

It is quite hard to get exact contour information from objects. Effective contour extraction must account differences in contrast and texture between the object and the backdrop, as well as the continuity of object borders. In real-life situations, the task is even harder because the background is more complex and the environment changes. When the goal is made easier by only needing to separate the foreground and background without needing to be very precise about the contours, the problem becomes easier to solve [3]. This simplified segmentation technique has shown to be useful in applications requiring real-time performance, such as video surveillance systems and autonomous driving.

In the past few years, many different algorithms have shown good success in separating the foreground from the background. Deep learning-based picture segmentation techniques, which depend on extensive annotated datasets, can attain excellent segmentation accuracy in intricate scenarios [4]. These approaches can better adapt to different visual contexts by learning hierarchical feature representations from a lot of training data. Still, these kinds of methods usually need a lot of computing power and a lot of labeled data, and their ability to generalize may still go down when they are used in situations that are new or very complicated.

Contour information can also be extracted by combining multiple traditional techniques for specific applications. For example, edge detection methods are commonly used to identify prominent boundaries within an image, while region-growing algorithms can expand these boundaries to cover entire object regions [5,6]. By integrating these approaches, more complete object contours can be obtained. Despite their effectiveness in controlled settings, traditional methods continue to face significant challenges in complex images, particularly under conditions involving occlusion or uneven illumination. These limitations indicate the need for further improvements in algorithm robustness and stability.

Our objective is to suggest a method that optimizes the performance of image recognition algorithms by leveraging extracted foreground and background information. Our objective is to create an approach that benefits from the distinct distinction between the foreground and background to enhance performance in image classification tasks.

2. Related work

Image classification has been an important part of computer vision for the last few decades, leading to a lot of study and development. The main goal of image classification is to give an input image a label based on what it looks like. This problem has several uses, such as object identification, self-driving cars, and finding images. Early picture classification algorithms mostly used classical machine learning techniques and features that were developed by hand. The Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) were well-known for their ability to pull features out of pictures. After then, classifiers like Support Vector Machines (SVMs) and k-nearest Neighbours (k-NN) [7,8] used these features to do the classification task. The advent of deep learning, especially convolutional Neural Networks (CNNs), has brought about a major change in the field of picture classification. CNNs independently obtain hierarchical feature representations from raw image pixels, eliminating the need for manual feature engineering. In this sense, AlexNet, which Krizhevsky et al. introduced in 2012, was a major step forward because it set a new standard for performance on the ImageNet dataset [2]. The success of AlexNet led to a lot of research on more complex and basic CNN designs. After AlexNet, many other important CNN architectures

DOI: 10.54254/2755-2721/2026.TJ31147

have been suggested. In 2014, Simonyan and Zisserman presented VGGNet, demonstrating that deeper networks with smaller convolutional filters might achieve enhanced performance [9]. Szegedy et al. developed GoogLeNet, also known as Inception, which employed an innovative inception module to capture multi-scale features, significantly decreasing the number of parameters relative to earlier architectures [10]. He et al. created ResNet in 2016 to solve the problem of vanishing gradients in very deep networks by adding residual connections. This new idea made it possible to create training networks with more than 100 layers, which pushed the limits of how well images could be classified [11]. DenseNet came after ResNet and included dense connections across layers. This made it easier to reuse features and improved the flow of gradients [12]. The transition from handmade features to deep learning has also been affected by the fact that powerful computers and vast datasets are now available. The ImageNet collection, which has millions of labeled photos in thousands of categories, has proven very important for the progress of image categorization. Moreover, the use of Graphics Processing Units (GPUs) has made it possible to train deep learning models on a massive scale in a short amount of time. Transfer learning is now a key method for classifying images, especially when there isn't much labeled data available. You can use the learned features to improve pretrained models on certain tasks, which means you don't need as much labeled data [13] and can use big datasets like ImageNet. This technique has proven effective across diverse domains, including remote sensing and medical imaging. Recent research has also looked into how to use attention mechanisms and transformers together to classify images. Dosovitskiy et al. created Vision Transformers (ViTs), which change the transformer design, which was first made for natural language processing, to work with image data. The competitive performance of transformers against CNNs [14] has shown how useful they may be for visual tasks. Self-supervised learning has gained popularity as a technique for training on huge amounts of unlabeled data. Contrastive Predictive Coding (CPC) and SimCLR are two methods that get useful feature representations by comparing positive and negative pairs of image regions. These techniques have shown promise in improving the resilience of image classification models and decreasing dependence on labeled data [15,16].Another big step forward is using generative models to produce more data and synthetic data. When there isn't a lot of data, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) can make realistic images that help with training [17,18]. Architectural advances have helped improve picture classification models, but optimization approaches and regularization methods have also proved very helpful. To make models work better and generalize better, techniques including batch normalization, dropout, and data augmentation have become routine [19,20]. Part of the continual advancement of picture categorization is building efficient models that may be used on peripheral devices. MobileNet and EfficientNet are two types of architectures that were designed with an emphasis on performance and computational efficiency trade-offs. This makes them good for real-time applications [21,22].

Image classification research has kept on developing over recent years. The examination of Vision Transformers (ViTs) along with their various forms has been a rather remarkable effort. In 2023, quite a few publications came out aimed at improving the effectiveness and performance of ViTs. For instance, Zhu et al. (2023) put forward a fresh architecture named Dynamic Vision Transformer (DyViT). This architecture can dynamically alter the model’s complexity based on the input image. Such an approach makes the trade-off between accuracy and computational cost more optimized [23]. Moreover, Wang et al. (2023) presented Hierarchical Vision Transformers (HVTs). These HVTs combine multi-scale features so as to capture contextual information more effectively. They have shown outstanding performance on benchmarks like ImageNet [24].

DOI: 10.54254/2755-2721/2026.TJ31147

3. Methodology - dataset preparation

Because of the limitations on computation, we utilized a part of the ImageNet dataset [2] for our experiments. This part includes 7 classes, with each class having 1300 images, making a total of 9100 images. At first, the images were processed by a depth estimation algorithm named DTP [25] to get depth maps, which is a key step for the later foreground-background segmentation. We assessed several depth estimation algorithms like DTP [25], Dinov2 [26], and Depth_Anything [27]. Even though there were notable differences in output quality at the pixel level among these algorithms, their overall effectiveness within our patch-based foreground–background segmentation approach was similar. The processing workflow goes like this:

• Depth Image Generation – We used the Dinov2 algorithm [28] to process the original photos and make depth images that matched them. This stage gives us the basic information we need to do the next step, which is to separate the foreground from the backdrop.

• Binarization – After getting the depth images, the Otsu method was used to binarize them. Otsu's approach is an adaptive thresholding technique that automatically finds the best threshold value depending on the image histogram. This lets you easily separate the foreground and background areas [29]. This phase makes the depth information easier to understand while keeping the important spatial structure.

• Image Region Division – After that, each binarized image was split into several little square areas called patches. This way of dividing things up is in line with the patch-based representation used in the Vision Transformer (ViT) framework, which makes it easy to deal with ViT-based models. The size of the patch was based on the resolution of the input image and the size of the patch that had already been set.

• Foreground and Background Classification – For each patch, the ratio of foreground pixels to the total number of pixels was found. A threshold of 0.5 was applied to give patches labels. In particular, patches were classed as foreground if they had a foreground pixel proportion of 0.5 or higher. The other patches were classified as background. This criterion offers a straightforward yet efficient method for distinguishing foreground from background at the patch level.

• Generation of Prevalence Maps – The algorithm doesn’t directly give out the segmented foreground and background. What it does is to produce a prevalence map. This map is in the form of a square image, and its dimensions are figured out by dividing the image’s pixel size by the patch size.

By going through these steps, we made good use of depth estimation techniques to divide images into the foreground and background areas. In this way, we got high-quality input data for the image processing tasks that would come after. This approach makes use of the advantages of different depth estimation algorithms. At the same time, it guarantees that it can work well with the ViT algorithm. As a result, the efficiency and effectiveness of the entire processing workflow are improved.Model Architecture

3.1. Preliminary experiments

Before formally introducing the model, we describe a simple experiment we conducted. The dataset used for this experiment is a subset of the COCO dataset [30]. For the background dataset, we used the bounding boxes from COCO's object detection annotations to crop objects of the same category from the original images [30]. Then, using the panoptic segmentation data, we extracted objects along their contours from the original images, filling the empty areas with black to generate the non-background dataset. Below is a diagram illustrating the datasets, as shown in Figure 1.

DOI: 10.54254/2755-2721/2026.TJ31147



Figure 1. Schematic diagram of the dataset

We did tests with two sets of data: one with seven categories and the other with fourteen. We used the Vision Transformer (ViT) and Swin Transformer V2 models, which we got from the official Timm library [14,31]. These are two different kinds of architecture: the traditional ViT structure and the pyramid-based structure.

Figure 2 shows that models that were trained on the non-background dataset have a better Top-1 accuracy and a faster convergence speed than models that were trained on the background dataset. Because there were only seven categories in the dataset, the change in TOP5 accuracy was not noticeable. But when employing the fourteen-category dataset, the non-background dataset did better with the Swin Transformer V2 in terms of TOP1 accuracy.

When it comes to the loss measures, the training loss and testing loss on the non-background dataset are always smaller than the loss on the background dataset. The loss function shows how far off the model's predictions are from the ground-truth labels. This means that lower loss values mean that the model is learning better. The lower loss on the non-background dataset indicates that taking out background information helps the model focus more directly on object-related attributes, which makes learning easier. Also, the smaller difference between training and testing loss means that the model trained on non-background data is better at generalizing, which means it works better when applied to samples it hasn't seen before. This makes us think about how we might be able to use this method to make a new neural network model.

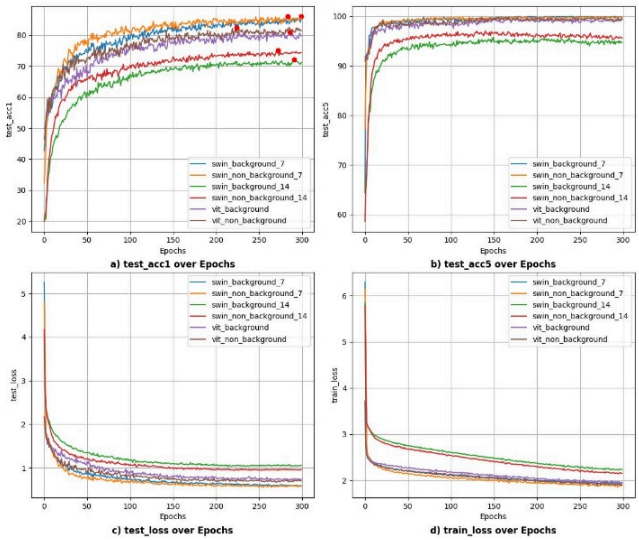

Figure 2. The demonstrates a two-part processing pipeline. In its upper part, an image is fed into a Vision Transformer Models (VTMs) model, which is subsequently followed by a linear layer. In the lower part, the original image is algorithmically divided into the foreground and background according to depth information. These components are separately processed via VTMs. Afterwards, the outputs from the VTMs are combined to decide the final classification

DOI: 10.54254/2755-2721/2026.TJ31147

3.2. Experiments

From first tests, we saw that the pyramid-structured Vision Transformer (ViT) works better than the old ViT architecture. Consequently, throughout the formal experimental phase, we shall utilize two distinct algorithms: BiFormer and Swin-Transformer [31]. This part gives a short overview of BiFormer and Swin-Transformer, as well as their benefits. The next part talks about BiFormer and Swin Transformer and their benefits.

• BiFormer - BiFormer is an improved vision transformer technique that uses a bidirectional transformer mechanism to better collect picture features, making it easier to analyze complicated images. It has the benefits of better handling long-range relationships and doing well at image classification and object identification tasks [32].

• Swin-Transformer - Swin-Transformer uses a sliding window system that makes it easy to analyze high-resolution photographs quickly. Its benefits include the ability to capture strong local features and improve model accuracy and generalization through hierarchical feature representation [31].

The datasets used for the studies will be split into two groups: one will include seven categories and the other will have thirty. Next, we will build a dual-path neural network structure using both the BiFormer algorithm and the Swin-Transformer technique. Figure 3 shows the whole structure of the suggested dual-path model.

• Input Data - Input data entails full images as well as the corresponding prevalence maps.

• Image Segmentation - The algorithms will use the prevalence maps to split the photos into the foreground and background. The blank areas will be filled in with black.

• Dual-Path Input - Two different BiFormer or Swin-Transformer neural networks will take the foreground and background pictures as input.

• Output Integration - The outputs from the two neural networks will be integrated immediately.

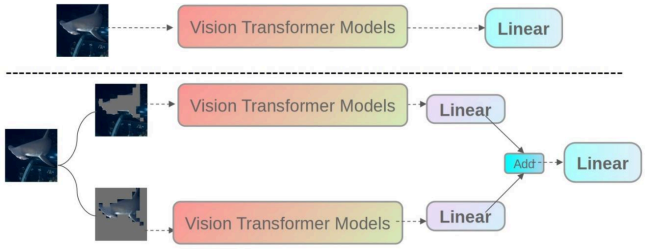

Figure 3. The demonstrates a two-part processing pipeline. In its upper part, an image is fed into a Vision Transformer Models (VTMs) model, which is subsequently followed by a linear layer. In the lower part, the original image is algorithmically divided into the foreground and background according to depth information. These components are separately processed via VTMs. Afterwards, the outputs from the VTMs are combined to decide the final classification

Moreover, we have meticulously crafted two comparative experiments:

• Dual-Path Full Image Input - Two full images will be input separately into the dual-path neural networks for processing.

• Shared Parameter Single-Path Network - In this case, the foreground as well as the background images are fed into a single-path neural network, with these two neural networks having shared parameters.

The following criteria will be employed to assess the performance of models::

DOI: 10.54254/2755-2721/2026.TJ31147

• Top-1 Accuracy - Top-1 Accuracy refers to the degree to which the model's most precise prediction aligns with the actual label. It indicates.

• Top-5 Accuracy - The percentage of instances in which the actual label is among the top five predictions of the model.

• Train Loss - Loss computed on the training dataset, which indicates the extent to which the model is learning during the training process.

• Test Loss - The loss calculated on the testing dataset, which indicates the model's ability to generalise to unobserved data.

The objective of this experimental design is to meticulously evaluate the efficacy of BiFormer and Swin-Transformer in various image processing tasks. It will capitalise on the dual-path neural network structure and the unique advantages of these devices to improve the accuracy and performance of the model.

4. Result and evaluation

4.1. Biformer-tiny model performance on cls30 and cls7 with varying parameters

The performance of the biformer_tiny model on the cls30 and cls7 datasets is demonstrated in figure 4 in terms of acc1, acc5, test loss, and train loss. The use of various parameters in the dual-path model is denoted by diff_parameter in these figures.

These observations show that simply increasing the number of paths in the biformer structure does not guarantee better performance. In fact, it can introduce challenges such as overfitting, increased training difficulty, and reduced generalization ability. One of the critical reasons for this is the inherent limitations in how the model handles the interaction between the foreground and background. In the dual-path ViT (Vision Transformer) architecture, the foreground and background are processed separately, with no direct interaction between them during the independent computation. This lack of interaction means that the model might miss important contextual information arising from the interplay between the foreground and background elements, leading to a less holistic understanding of the scene.

The model's ability to generalize may be negatively impacted if it becomes excessively specialized to specific processing paths, resulting in the amplification of noise or extraneous characteristics in both the foreground and background. Separating these channels may also make the model less effective in finding relationships between different parts of an image, especially when the meaning of something depends on information that is shared across the foreground and background. Because of this, the model could have a hard time learning a well-balanced representation. This makes the training process harder and increases the chance of overfitting.

I The experimental results further demonstrate that when the training set has a restricted number of classes, the performance disparity between the single-path and dual-path models is minimal. But as the number of classes increases, the original single-path model starts to do better than the dual-path model. This suggests that the dual-path approach has a harder time scaling when tasks get more complicated. The test loss curve, which goes down at first and then up again, supports this observation even more. This shows that the dual-path model's learning process is unstable. One probable reason is that the fusion approach utilized is rather simple: at the end of the classification process, the features from the foreground and background are added together immediately. This kind of fusion method might make it harder for the two feature streams to interact in a meaningful way, which would limit the overall representational capacity and lower performance.

DOI: 10.54254/2755-2721/2026.TJ31147

• Insufficient Training Data - The lack of training data is another issue. The training data has a limited number of classes and samples. This situation could result in unstable performance when conducting training and testing. It is especially manifested as the rising trend of test loss.

• Larger Parameter Size in Dual-Path Model - Dual-path models have a larger parameter size. The number of parameters in this model is doubled, which raises the difficulty of training and decelerates the speed at which the loss is reduced.

• Simple Model Fusion Mechanism - Simple Model Fusion Mechanism – The present straightforward model fusion method involves directly summing the outputs of the foreground and background at the final class output stage. This approach might lead to an insufficient interaction between the foreground and background, thereby having a negative impact on the model's performance.

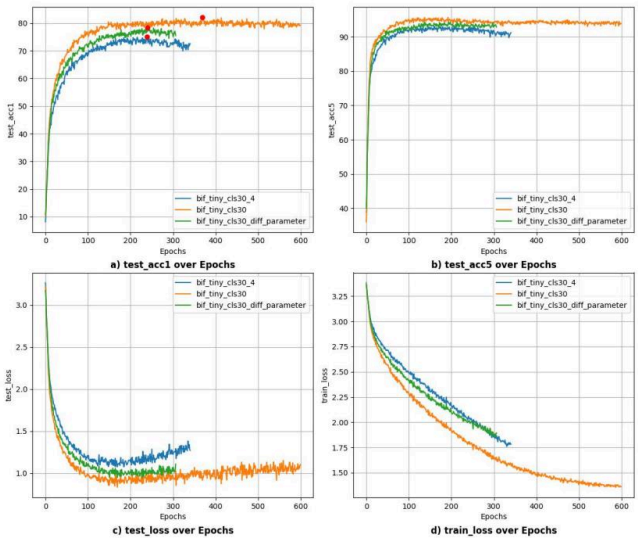

Figure 4. Performance of the biformer_tiny model on cls30 and cls7 datasets. (a) Top-1 accuracy (acc1): The performance of single-path and dual-path models is similar with fewer classes, but the single-path model performs better with more classes. (b) Top-5 accuracy (acc5): A similar trend as in acc1, with the single-path model outperforming as the number of classes increases. (c) Test loss: Initially decreases and then increases, possibly due to limited data. (d) Train loss: The dual-path model shows a slower decrease in loss, likely due to its larger parameter size

4.2. Path count impact on biformer performance

As shown in Figure 5, the experiments demonstrate the performance of single-path, dual-path, and quad-path biformer structures on the dataset. The results indicate that increasing the number of paths in the biformer structure does not necessarily lead to improved performance. On the contrary, a higher number of paths can cause the algorithm to overfit more efficiently, and the quad-path model proves to be more challenging to train compared to the dual-path model.

• Performance Degradation with Increased Paths - The experiments indicate that the multi-path biformer does not produce superior results. Performance appears to be impeded rather than improved by the complication imposed by multiple paths.

• Multi-Path Model Overfitting - The likelihood of overfitting increases as the number of paths increases. This is most likely the result of the model's increased capacity, which can result in the model suiting the noise in the training data rather than successfully generalizing to new data.

DOI: 10.54254/2755-2721/2026.TJ31147

• Training Difficulty - The quad-path biformer is observed to be more challenging to train than the dual-path model. The complexity of the interactions and the need for parameter tuning in a model with more paths are likely the reasons for the increased difficulty.

The main insights into the impact of path count on model performance are revealed by the experiments conducted on single-path, dual-path, and quad-path biformer architectures. The results contradict the anticipation that the efficacy could be enhanced by the inclusion of additional paths. The performance of the biformer structure is not necessarily improved by the increase in the number of paths; in fact, it may contribute to performance degradation.

The model is more likely to overfit as the number of routes goes up. This is particularly clear in the quad-path setup. The diminished generalization performance is probably due to the increased model capacity, which promotes the acquisition of noise instead of significant patterns. Also, the quad-path model is harder to optimize than the dual-path model because the routes interact in more complicated ways and are more sensitive to changes in the parameters. These results show how important it is to properly balance model complexity. Just adding more routes doesn't always make things better; in fact, it might make training less stable and lead to overfitting.

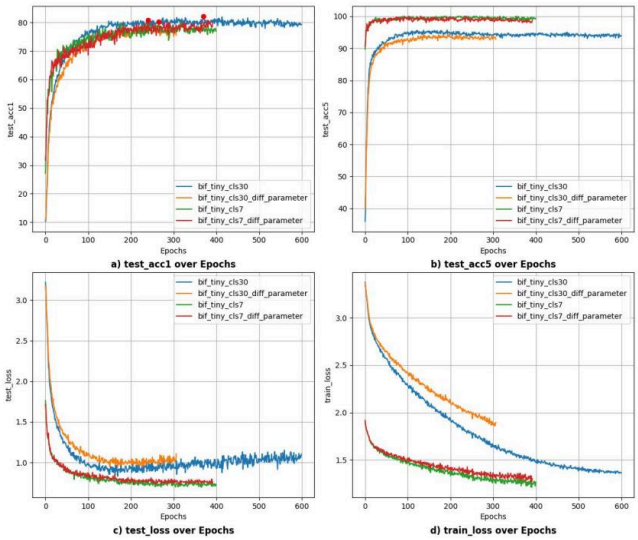

Figure 5. The graphic compares single-path, dual-path, and quad-path BiFormer models. It shows that adding more routes doesn't always make things work better. As the number of paths increases, performance degradation becomes apparent, with the quad-path model showing the most significant reduction. This model is also harder to train, probably because it is more complicated because routes interact with each other in ways that make overfitting worse instead of better

4.3. Comparison of parameter-sharing and non-parameter-sharing neural networks

In the last trials, a dual-path Vision Transformer (ViT) architecture was used. Two distinct design techniques were tested: one where the two paths shared parameters and another where each path kept its own parameters. Figure 6 shows how these two setups performed compared to each other throughout training. It shows that they learned in very different ways.

The top-1 accuracy (Acc1) for the parameter-sharing model went up substantially in the beginning of training and then leveled off quite quickly. But after this improvement, accuracy started to go down as training went on. This trend shows that while sharing parameters can speed up convergence, it might also make it harder for the model to generalize, which could cause performance to drop once the best point is reached. The model that didn't have common parameters,

DOI: 10.54254/2755-2721/2026.TJ31147

on the other hand, showed a more slow but steady increase in accuracy. The non-shared configuration kept getting better even after the shared-parameter model started to go down. This shows that it was more stable throughout later training phases.

The examination of test loss corroborates these findings. In the shared-parameter mode, the test loss went up toward the conclusion of training, which was the same time when the accuracy went down. The non-shared model, on the other hand, had a steady or slightly decreasing test loss, which means it was more resilient and less likely to overfit. In general, our results show that there is a clear trade-off between speedier convergence and long-term performance stability in dual-path ViT designs. Although parameter sharing may yield initial performance advantages, the non-shared setup eventually ensures greater reliability and enduring accuracy, rendering it the preferred option where generalization and training stability are paramount.

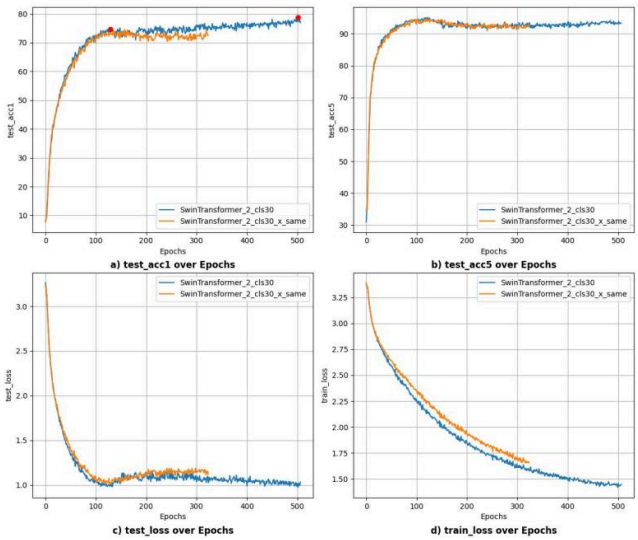

Figure 6. Figure (a) compares Acc1 (Top-1 accuracy). It shows that the parameter-sharing model converges quicker at first, but its performance drops once it reaches a peak. The non-parameter-sharing model, on the other hand, shows more stable behavior over time. Figure (b) shows a similar pattern for Acc5 (Top-5 accuracy), where the non-parameter-sharing model always has superior accuracy as training goes on. Figure (c) shows the test loss, which goes up late in the process for the parameter-sharing model. This means that the model is less likely to generalize. At the same time, the non-parameter-sharing model exhibits test loss that stays the same or gets a little better. Figure (d) shows the training loss, and the parameter-sharing model has lower values. However, this doesn't mean that the model will do better on tests, which could mean that it is overfitting

4.4. Limitations

One major limitation of this study is the relatively small scale of the dataset. The experiments were conducted on datasets containing only seven and thirty classes, with approximately 7,000 to 30,000 images in total. Compared with large-scale benchmarks such as ImageNet, which include millions of images across diverse categories, this limited data scale may constrain the generalizability of the findings and affect overall model performance.

Another limitation arises from the model architectures used in the experiments. Smaller variants of BiFormer and Swin Transformer were adopted, featuring fewer parameters than their larger versions. While these lightweight models are computationally efficient, they may not fully capture

DOI: 10.54254/2755-2721/2026.TJ31147

the advantages of the dual-pathway design. The reduced model capacity could limit performance, particularly when extending to more complex tasks or larger datasets.

In conclusion, the robustness and scalability of our findings may have been influenced by the limited size of the dataset and the use of simpler, parameter-constrained models. In order to further validate and improve the performance of the proposed methods, future research should explore the use of larger datasets and more intricate model architectures.

4.5. Reflection and outlook

We have learned that, even if multi-path models seem like a good idea in theory, the best way to use them in practice is not just to add more pathways. A lot of experiments led to this realization. The basic additive fusion technique used for proximal and background interaction does not fully fulfill the possibilities of the dual-path structure.

We can make the interaction between the foreground and backdrop in future models even better by looking at a number of different technical approaches. First, the current model uses a rather simple additive fusion method, which could limit the full potential of foreground-background interaction. You could make a more advanced fusion mechanism, such adaptive weighting or attention-based fusion, that changes the contribution of each path based on the content and context of the input data. This would help the model better tell the difference between important characteristics in the front and background, which would make it work better.

Second, various ways to figure out depth could be employed to make it easier to tell the difference between the foreground and background parts. For example, adding binocular or multi-view depth data to the model could assist the algorithm better recognize the difference between things that are close by and the background by giving it more complete spatial information. This could be quite helpful when you require depth cues to figure out how the scene is put together.

Examining the potential for multi-scale feature extraction and fusion may enhance the interaction between the foreground and background. The model was able to get a more accurate and detailed representation of the landscape by recording elements in more than one dimension. This helped it figure out how the different pieces of the picture fit together.

Finally, the model's ability to tell the difference between the foreground and background may be better if it uses temporal information from video sequences when it makes sense to do so. Temporal consistency and motion signals could give the model more information, which could help it better monitor and separate moving objects in a scene.

5. Result and evaluation

Our research shows that the performance of image classification can be improved by using foreground and background segmentation. At first, we found that models trained on datasets without background got better accuracy and converged faster. However, adding more paths like dual-path or quad-path structures didn’t always make the model perform better. It often led to overfitting and made training harder. The simple way of fusing models, which treats the foreground and background as separate images, didn’t optimize the algorithm’s performance either. In the future, we need to focus on creating better fusion techniques so that foreground and peripheral information can interact effectively. Also, we have to find ways to overcome the challenges of accurately defining class contours and dealing with the increased complexity of models to improve image recognition algorithms.

DOI: 10.54254/2755-2721/2026.TJ31147

References

[1] D. Marr, Vision: A computational investigation into the human representation and processing of visual information. Mit Press, 1982.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks, ” Communications of the ACM, vol. 60, pp. 84–90, 05 2012.

[3] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context, ” International Journal of Computer Vision, vol. 81, pp. 2–23, 12 2007.

[4] K. He, G. Gkioxari, P. Doll´ar, and R. Girshick, “Mask r-cnn, ” in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp.2980–2988.

[5] J. Canny, “A computational approach to edge detection, ” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[6] R. Adams and L. Bischof, “Seeded region growing, ” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, pp. 641–647, 1994.

[7] D. Lowe, “Object recognition from local scale-invariant features, ” in Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, 1999, pp. 1150–1157 vol.2.

[8] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection, ” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 1, 2005, pp. 886–893 vol. 1.

[9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition, ” 2015. [Online]. Available: https: //arxiv.org/abs/1409.1556

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions, ” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition, ” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[12] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks, ” 2018. [Online]. Available: https: //arxiv.org/abs/1608.06993

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” 2014. [Online]. Available: https: //arxiv.org/abs/1411.1792

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale, ” 2021. [Online]. Available: https: //arxiv.org/abs/2010.11929

[15] O. J. H´enaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, “Data-efficient image recognition with contrastive predictive coding, ” 2020. [Online]. Available: https: //arxiv.org/abs/1905.09272

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations, ” 2020. [Online]. Available: https: //arxiv.org/abs/2002.05709

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks, ” 2014. [Online]. Available: https: //arxiv.org/abs/1406.2661

[18] D. P. Kingma and M. Welling, “Auto-encoding variational bayes, ” 2022. [Online]. Available: https: //arxiv.org/abs/1312.6114

[19] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift, ” 2015. [Online]. Available: https: //arxiv.org/abs/1502.03167

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting, ” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 01 2014.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications, ” 2017. [Online]. Available: https: //arxiv.org/abs/1704.04861

[22] M. Tan and Q. V. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks, ” 2020. [Online]. Available: https: //arxiv.org/abs/1905.11946

[23] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, “Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition, ” 2021. [Online]. Available: https: //arxiv.org/abs/2105.15075

[24] C. Ryali, Y.-T. Hu, D. Bolya, C. Wei, H. Fan, P.-Y. Huang, V. Aggarwal, A. Chowdhury, O. Poursaeed, J. Hoffman, J. Malik, Y. Li, and C. Feichtenhofer, “Hiera: A hierarchical vision transformer without the bells-and-whistles, ” 2023. [Online]. Available: https: //arxiv.org/abs/2306.00989

DOI: 10.54254/2755-2721/2026.TJ31147

[25] R. Ranftl, A. Bochkovskiy, and V. Koltun, “Vision transformers for dense prediction, ” 2021. [Online]. Available: https: //arxiv.org/abs/2103.13413
[26] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, “Dinov2: Learning robust visual features without supervision, ” 2024. [Online]. Available: https: //arxiv.org/abs/2304.07193
[27] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, “Depth anything: Unleashing the power of large-scale unlabeled data, ” 2024. [Online]. Available: https: //arxiv.org/abs/2401.10891
[28] X. Wang, L. Xie, C. Dong, and Y. Shan, “Real-esrgan: Training real-world blind super-resolution with pure synthetic data, ” in 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 1905–1914.
[29] N. Otsu, “A threshold selection method from gray-level histograms, ” IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, 1979.
[30] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Doll´ar, “Microsoft coco: Common objects in context, ” 2015. [Online]. Available: https: //arxiv.org/abs/1405.0312
[31] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, “Swin transformer v2: Scaling up capacity and resolution, ” 2022. [Online]. Available: https: //arxiv.org/abs/2111.09883
[32] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. Lau, “Biformer: Vision transformer with bi-level routing attention, ” 2023. [Online]. Available: https: //arxiv.org/abs/2303.0881