# A Survey: Research Progress of Feature Fusion Technology

Weiqi Wang[1,2,3], Feilong Bao[1,2,3(✉)], Zhecong Xing[3], Zhe Lian[4]

[1] National & Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Hohhot 010000, China
[2] Inner Mongolia Key Laboratory of Mongolian Information Processing Technology College of Computer Science, Inner Mongolia University, Hohhot 010000, China
[3] College of Computer Science, lnner Mongolia University, Hohhot 010000, China
[4] School of Computer Science and Technology, Inner Mongolia Normal University, Hohhot 010022, China
csfeilong@imu.edu.cn

**Abstract.** Feature fusion techniques represent a critical research content in the domain of deep learning, aiming to concatenate feature information from diverse sources or varying levels to generate more comprehensive and accurate representations. This technology is extensively employed in downstream tasks that necessitate rich target representations, such as image classification, semantic segmentation, and object detection. Over recent years, under the impetus of advancements in deep learning technologies, we have witnessed rapid progress in feature fusion techniques and their profound impact on the entire computer vision field. This paper takes an technique evolutionary perspective to comprehensively summarize the innovative contributions of feature fusion technology within four cutting-edge domains: Convolutional Neural Network (CNN), Vision Transformer (ViT), Graph Convolutional Network (GCN) and Neural Architecture Search (NAS). We provide a detailed introduction to the specific implementation process of each technology, and analytically explores the pivotal roles played by the concept of feature fusion in each of these technologies through different viewpoints. Finally, we provides a systematic overview of the mechanisms behind several classical methods and arrange the open-source code links, and we performance evaluation was conducted on several classic methods.

**Keywords:** Deep learning, Feature fusion, Convolutional neural network, Vision transformer, Neural architecture search, Graph convolutional network.

## 1 Introduction

With the advent of the big data era and the rapid advancement of computer technologies, feature fusion techniques have increasingly found broad applications across various domains including Artificial Intelligence, Machine Learning, Computer Vision, and Natural Language Processing. Serving as a vital tool for information

processing and data analytics, these techniques effectively amalgamate feature information from disparate sources and with differing attributes, thereby enhancing the recognition and classification capabilities of systems. This improvement, in turn, propels the development of more intelligent and precise information processing and applications.

The research on feature fusion technology began at the end of the last century, and with the continuous development of computer vision and pattern recognition, its research depth and breadth have gradually expanded. Early feature fusion methods were mainly based on simple weighted summation or feature connections, such as Hypercolumns, ION, YOLO v2 [1, 2, 3] etc. Other algorithms SPPnet and Inception use multi-scale convolution kernels to extract features from different receptive fields [4, 5], and merge the extracted features to achieve feature fusion. Although these methods are simple and easy to implement, and have been successful in early computer vision tasks, they are no longer applicable as the complexity of data and models increases, as they ignore the inherent correlation and complementarity between different features, resulting in limited fusion effects. Therefore, researchers are committed to finding more effective feature fusion methods to adapt to higher dimensional and more complex task requirements [6].

As research delved deeper, scholars began exploring more sophisticated feature fusion techniques. Accompanied by the rapid advancements in deep learning technologies, the study of feature fusion methods has seen remarkable progress, the most typical example is ResNet proposed by He Kaiming *et al*. [7], has amassed over 200,000 citations in the field of artificial intelligence. The residual connections in ResNet exemplify the rudimentary manifestation of the feature fusion paradigm. Building upon this successful groundwork, researchers have embarked on extensive investigations, integrating feature fusion technology with Convolutional Neural Network (CNN), Vision Transformer (ViT), Graph Convolutional Network (GCN), and Neural Architecture Search (NAS) techniques [8, 9, 10, 11]. It provides powerful performance improvements for various complex tasks. These technologies complement and promote each other, jointly driving continuous progress in the field of artificial intelligence. With the continuous development of technology and the expansion of application scenarios, feature fusion technology will play a more important role in the future, providing more efficient and accurate solutions for various complex tasks.

This paper introduces the combination of feature fusion technology with CNN, ViT, NAS, and GCN in the Section 2, Section 3, Section 4, and Section 5. Several key representative works are listed for each technology, and the role of feature fusion ideas is explained. In the Section 6, each method mechanism is summarized and links to open source code for different technologies are organized. Summarize the entire article in Section 7.

## 2 CNN and Feature Fusion Technology

In well-designed CNN architectures, the notion of feature fusion has long been ingrained, exemplified by two prominent cases: Residual connections of ResNet [7],

where input feature maps are fused directly with the output of the last convolutional layer via a shortcut connection, with their sum serving as the output of the residual block. Another example is the feed-forward connections of DenseNet [12], which implement fusion functionality through concatenative depth-wise aggregation. These fusion strategies in both networks emerged during the nascent stage of feature fusion thinking and primarily involved straightforward addition or concatenation. Over time, the development and refinement of feature fusion techniques have led researchers to broaden their perspectives on CNN design. HRNet and CEDNet [13, 14], for instance, incorporate more sophisticated feature fusion methodologies, demonstrating formidable capabilities as a result. These networks reflect the evolution in which advanced feature fusion techniques are being increasingly integrated into CNN architectures.

### 2.1 ResNet

In 2015, the team led by Kaiming He proposed ResNet [7], still one of the most popular backbone networks in the field of computer vision. The central idea of ResNet revolves around residual learning, where residual blocks are ingeniously designed to constitute the fundamental building blocks for constructing deep networks, as depicted in **Fig. 1**.

Within the residual block, the input feature map undergoes direct fusion with the output of the final convolutional layer via a residual connection, which subsequently becomes the output of block. Through this residual fusion concept, the degradation problem associated with vanishing gradients in progressively deeper layers is alleviated, thus allowing the network to reliably extract high-level semantic features from images.

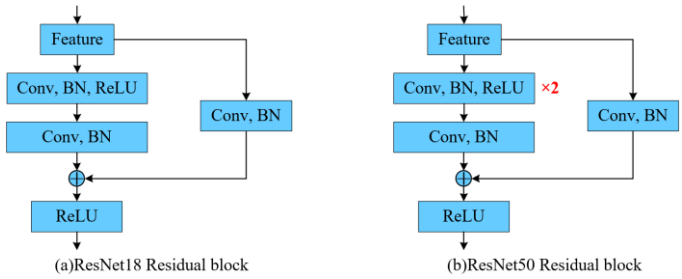


**Fig. 1.** Residual block structure diagram.

### 2.2 DenseNet

The design inspiration for DenseNet comes from ResNet [12], which builds a network based on dense blocks, as shown in **Fig. 2**.

The primary distinction between dense blocks and residual blocks lies in the fact that dense blocks employ a more interconnected feedforward mechanism where any layer feature map is concatenated to all subsequent layers. This approach mitigates the issue of vanishing gradients, optimizes the feature propagation pathways, and promotes

feature reuse. Unlike the approach of element-wise summing of ResNet, DenseNet utilizes depthwise concatenation for feature fusion, which offers the advantage that when feature maps from two layers exhibit entirely different distributions, it does not hinder the flow of information through the network.

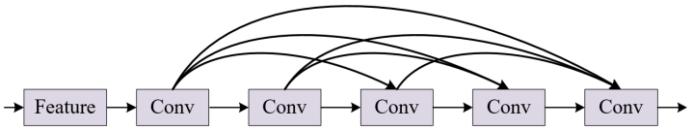


**Fig. 2.** Dense block structure diagram.

### 2.3 HRNet

Wang *et al*. reviewed previous outstanding frameworks for backbone network architectures [13], encompassing AlexNet [15], ResNet [7], DenseNet [12], VGGNet [16], and GoogleNet [5], and observed that these networks adhere to the design principles established by LeNet [17], which involves progressively reducing the spatial dimensions of feature maps, cascading convolutions from high to low resolutions, and generating lower resolution representations for further processing in downstream tasks. However, for position-sensitive tasks such as semantic segmentation, human pose estimation, and object detection, high-resolution representations are actually required. Consequently, Wang *et al*. introduced the High-Resolution Network (HRNet). The HRNet begins with a high-resolution convolutional stream and incrementally fusions this high-resolution stream with progressively lower resolution streams, connecting and merging parallel multi-resolution streams throughout the process. The purpose of HRNet is to maintain high-resolution representations consistently throughout the entire computational pipeline, thereby catering to the demands of tasks that require fine-grained positional information.

Most fusion approaches aggregate high-resolution low-level features with upsampled high-resolution representations derived from low-resolution representations. However, HRNet adopts a distinct strategy by repeatedly engaging in multi-resolution fusion to enhance the interoperability between high-resolution and low-resolution representations, enabling the improvement of high-resolution representations with the aid of low-resolution ones and vice versa. Consequently, this tactic ensures that all high-resolution to low-resolution representations carry strong semantic information, thereby boosting the performance and generalization capacity of the model.

### 2.4 CEDNet

Zhang *et al*. define the feature fusion time as the ratio of the parameters of the subnetwork preceding the first fusion module to the total network parameters [14], where a smaller ratio indicates an earlier fusion stage. They analyze mainstream multi-scale fusion methods such as Feature Pyramid Network (FPN) and (Bi-directional

Feature Pyramid Network) BiFPN [18, 19], pointing out that the feature fusion occurs relatively late in these networks since they allocate a substantial portion of computational resources to the classification backbone for extracting initial multi-scale features. For instance, in an FPN built upon ConvNeXt [8], the feature fusion time constitutes only 91.7%. There also exist methods that integrate multi-scale fusion at an earlier stage, with representative work being HRNet, which has a feature fusion time of 2.7%. However, despite its early fusion strategy, HRNet delays the generation of advanced (low-resolution) features with strong semantic information until later stages. This limitation hampers the role of the model in guiding the learning of low-level (high-resolution) features, which are critically important for tasks such as object detection.

To address the aforementioned issues, Zhang *et al*. propose a Cascade Encoder-Decoder Network, denoted CEDNet, which fusions multiple cascaded stages starting from a stem module that extracts initial high-resolution features. In CEDNet, a common encoder-decoder architecture is shared across all stages to generate multi-scale features. The fundamental building block of CEDNet is the CED unit, realized by a spatial feature interaction token mixer and a two-layer Multi-Layer Perceptron (MLP) for channel-wise feature interactions. The token mixer can adopt various existing designs, such as 3×3 convolutions in ResNet, 7×7 depthwise convolutions in ConvNeXt, or local window self-attention in the Swin Transformer [20]. The CEDNet explores three classical encoder-decoder structures: FPN [18], Hourglass [21], and U-Net [22]. It demonstrates superior performance when implemented on top of FPN, thereby providing further compelling evidence for the efficacy of feature fusion techniques.

## 3 ViT and Feature Fusion Technology

ViT is a Transformer like network model proposed by the Google team in 2020, which successfully applies Transformer, an architecture from the NLP field, to image classification tasks for the first time [23]. In the same year, FPT leveraged components from ViT to enhance feature fusion techniques [24], thus inspiring subsequent work. Subsequently, PVT emerged [25], being the first to seamlessly integrate a feature fusion structure with ViT, and following this innovation, classic hierarchical ViT architectures have been employed in the field of computer vision, demonstrating remarkable capabilities.

### 3.1 FPT

Zhang *et al*. were the first to attempt leveraging ViT technology to enhance feature fusion performance [24], combining the ideas of Transformer architecture and feature pyramid fusion, proposing an inter-spatial and cross-scale feature interaction fusion structure termed FPT (Feature Pyramid Transformer).

Unlike simple feature pyramid fusion structures, FPT adopts the self attention mechanism of Transformer, allowing for global information exchange between features at different levels, thus better integrating contextual information and multi-scale features. The FPT sequentially feeds multi-scale features generated by the backbone

network into three ViT-based components: ST (Self-Transformer), GT (Grounding Transformer), and RT (Rendering Transformer). The ST aims to capture co-existing target features within a feature map, adopting the classical non-local interactions proposed in literature [26], where both input and output maintain the same scale ratio. The GT performs non-local interactions in a top-down manner, grounding the concepts from high-level feature maps onto lower-level pixels, resulting in an output that has the same scale ratio as the lower-level feature maps. The RT operates in a bottom-up fashion, aiming to present higher-level concepts by incorporating visual attributes into low-level pixels, with its output having the same scale ratio as the higher-level feature maps. Through these three components, FPT transforms any feature pyramid fusion structure into one of equal size but enriched contextually.

### 3.2 PVT

Wang *et al*. propose a pure ViT based backbone network [25], named PVT (Pyramid Vision Transformer), which constitutes the first entirely convolution-free attempt at a target detection framework. The core idea lies in the introduction of a progressively shrinking pyramid that utilizes simple yet effective feature fusion techniques to flexibly learn multi-scale and high-resolution features. As depicted in **Fig. 3**, the block embedding component in PVT integrates and consolidates features originating from diverse data sources by fusing different feature maps, thereby enabling the model to more efficiently capture essential information about target objects. In the decoder of the PVT architecture, the principle of feature fusion is similarly employed. The PVT model adopts a bottom-up decoding approach, progressively merging lower-level features upwards, allowing for high-level features to better express the semantic information of target objects, thus enhancing the ability of model to accurately identify and localize such objects. Crucially, the design of PVT incorporates a progressive shrinking pyramid that generates four distinct scales of feature maps, which can be effortlessly integrated with feature fusion techniques like FPN, thereby yielding richer feature representations. Due to its outstanding performance, the PVT model achieved the TOP1 accuracy in the target detection methods of that year.

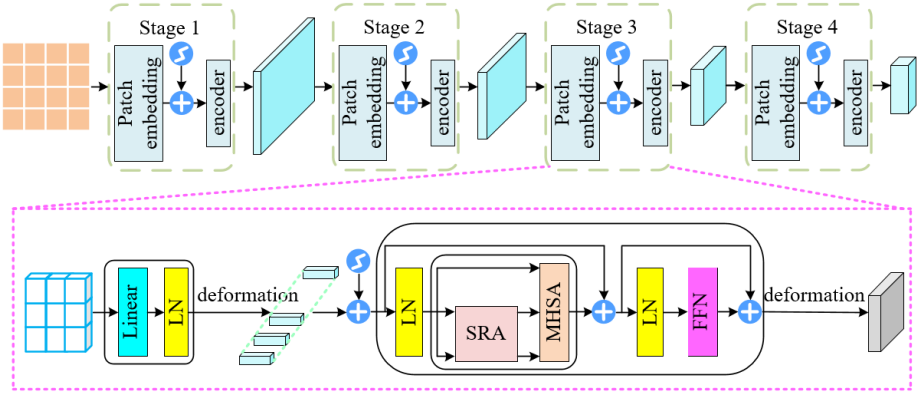


**Fig. 3.** PVT network architecture diagram.

## 4 NAS and Feature Fusion Technology

Since the advent of the feature fusion structure FPN, NAS technology was among the earliest to be combined with it, giving rise to Auto-FPN and NAS-FPN [27, 28]. NAS, which automatically optimizes neural network architectures, transcends the confines of traditional manual design paradigms. Upon integration with feature fusion techniques, NAS not only seeks optimal solutions at the macroscopic structural level but also refines the micro-level feature representations. Thus, it gives rise to more potent hybrid feature representations.

### 4.1 Auto-FPN

With the advancements in NAS techniques in the field of image classification, feature fusion technology has also explored NAS applications. Xu *et al*. posit that higher-level neurons tend to be sensitive to overall object shapes [27], whereas other neurons are more likely to be activated by local textures. To accommodate all possible connection patterns, they propose an Automatic Feature Pyramid Network (Auto-FPN), devising an automated fusion module (represented by the green dashed box in **Fig. 4**) embedded within a fully connected search space (indicated by the purple dashed box in **Fig. 4**). This module employs a variety of operations, such as dilated convolutions, residual connections, and depthwise separable convolutions, to provide ample spatial awareness and receptive field coverage for the feature fusion process. By doing so, it searches for the optimal spatial arrangements and architectural configurations.

Regarding neural architecture search methods, approaches based on Evolutionary Algorithms and Reinforcement Learning typically necessitate extensive retraining and evaluation of candidate architectures [29, 30], resulting in high computational demands even for low-resolution image classification tasks. Inspired by the differentiable formulation in NAS [31], Auto-FPN adopts a continuous relaxation of the discrete structures of its two modules. Subsequently, it conducts architecture search directly on high-resolution images of 800×800 pixels using stochastic gradient descent, thereby addressing the computational complexity issue.

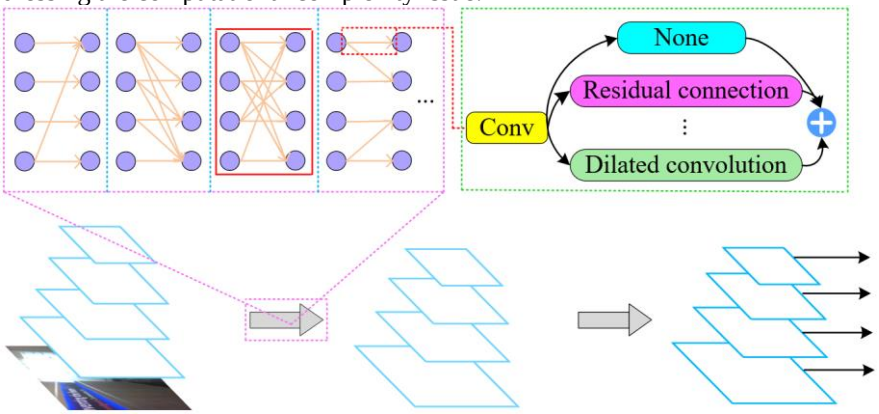


**Fig. 4.** Auto-FPN network architecture diagram.

### 4.2 NAS-FPN

Ghiasi *et al*. consider the application of NAS from another angle [28], an atomic architecture has been discovered in a new scalable search space that covers all cross scale connections, has the same input and output feature levels, and can be repeatedly applied to overcome the large search space of pyramid architecture, making modular search space possible. Based on this discovery, they proposed the Neural Architecture Search Feature Pyramid Network (NAS-FPN), the structure of which is depicted in **Fig. 5**.

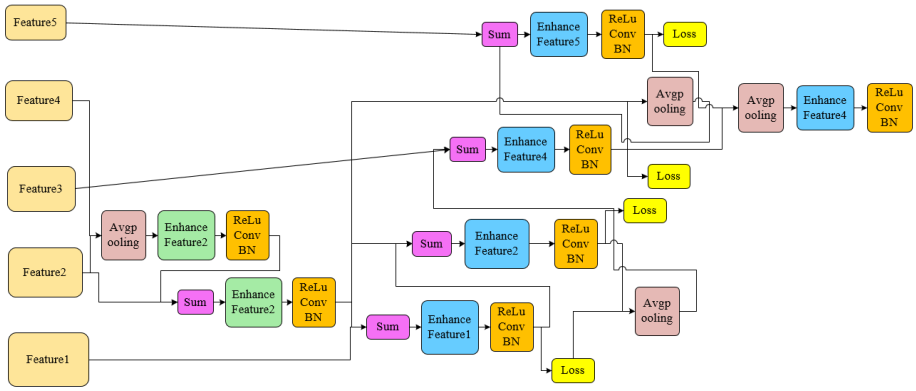


**Fig. 5.** NAS-FPN network architecture diagram.

In previous approaches, the rationale for feature fusion has been consistent, with the necessity to integrate features across different scales. Therefore, Ghiasi *et al*. devised a Merge Unit Module, where the entire NAS-FPN is composed of such Merge Units, the configuration of which is illustrated in **Fig. 6**. The construction of these Merge Units is determined by an RNN [32], which selects any two candidate feature layers with differing resolutions for binary operations, thereby combining them into a new feature layer. Specifically, this process comprises the following five steps:

(1)Select one feature from the pool of candidate features.

(2)Without replacement, select another feature from the candidate features.

(3)Determine the resolution of the output feature layer.

(4)Choose a binary operation (either global pooling or summation, which does not introduce any additional trainable parameters) to combine the two features selected in steps (1) and (2), thereby generating a feature layer with the resolution chosen in step (3).

(5)The newly generated feature layer is appended to the existing list of input candidate features and becomes a new candidate for the next Merge Unit.

During the architecture search phase, there may exist multiple candidate features sharing identical resolutions. To mitigate computational overhead within the architecture and to avoid selecting features with larger strides for intermediate Merge Units during Step 3, five Merge Units are designed to output a Feature Pyramid. This strategy ensures that redundancy is reduced and unnecessary upscaling or downscaling operations are minimized, thereby constructing an efficient feature hierarchy.

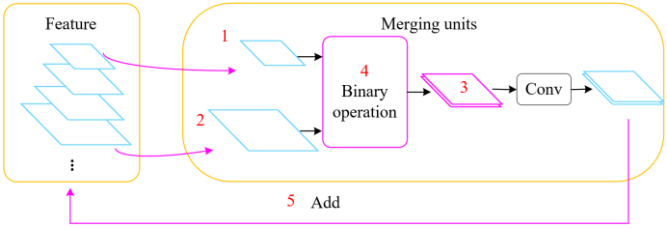


**Fig. 6.** Schematic diagram of merging units.

## 5 GCN and Feature Fusion Technology

In the context of GNN, a typical application of feature fusion is the integration of signals across different graph depths to bolster the expressive power and understanding of complex graph structures of model. Representative works that combine GCN with feature fusion techniques include GraphFPN and LFPN [33, 34].

### 5.1 GraphFPN

Zhao *et al*. argue that some advanced feature fusion technologies [33], such as BiFPN and RFPN [19, 35], introduce learnable weights and still perform feature interactions on neural network spaces and scales with fixed topologies, whereas the intrinsic structures of different images vary significantly. To address this, they introduce graph convolutional technology into the feature fusion method, thus proposing the Graph-Feature Pyramid Network (Graph-FPN), which dynamically models the part-whole hierarchies akin to human visual perception in different images. This network structure, depicted in **Fig.** 7, adaptively captures the part-whole relationships within scenes according to their specific structures.

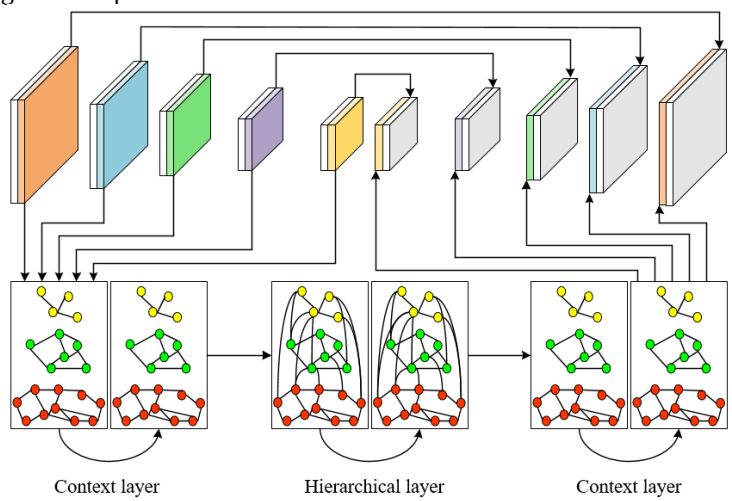


**Fig. 7.** Graph-FPN network architecture diagram.

Graph-FPN defines a multi-level superpixel hierarchy representing the inherent image structure, where each level consists of a set of non-overlapping superpixels that delineate the segmentation of the input image. Features are extracted from the same hierarchical segmentation across all levels for the input image. Consequently, superpixels at adjacent levels in the hierarchy bear close relationships; each superpixel at a higher level is a union of superpixels from the lower level. This one-to-many correspondence between superpixels across two adjacent levels inherently defines the part-whole relationship. Graph-FPN inherits its structure from this superpixel hierarchy, incorporating attention mechanisms akin to those in SENet and DAN within its contextual graph layer and hierarchical graph layer embeddings [36, 37]. This enables enhanced feature interaction not only within the same scale but also across different scales, thereby promoting a more comprehensive and adaptive understanding of the part-whole relationships within the image content.

### 5.2 LFPN

Similarly capitalizing on the idea of graph convolution, Xie *et al*. [34], building upon the design of GloRe [38], take feature maps into a latent space and utilize graph convolutions on the projected space to model long-range dependencies, thus proposing the Latent Feature Pyramid Network (LFPN), whose structure is illustrated in **Fig. 8**.

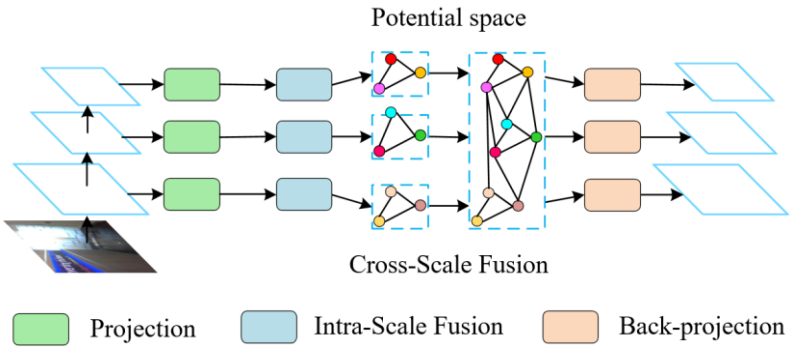


**Fig. 8.** LFPN network architecture diagram.

LFPN mainly consists of four parts:

(1)Projection: In grid space, feature fusion tends to lack long-range dependencies. Consequently, LFPN learns a projection matrix to map features onto a lower-dimensional latent space.

(2)Intra-Scale Fusion: Aimed at capturing the intra-scale relationships among different feature vectors, LFPN employs graph convolutional layers to learn these relations in an end-to-end manner, initializing and updating the adjacency and parameter matrices randomly.

(3)Cross-Scale Fusion: After intra-scale fusion, the combined features are linked together and fed through graph convolutional layers to capture long-range dependencies among multi-level features.

(4)Back-projection: Via an inverse projection matrix, the features are back-projected from the latent space back to the original grid space.

By conducting intra-scale and cross-scale feature fusion in the latent space, LFPN models long-range dependencies both within and across scales, thereby enriching the feature representation and enhancing its discriminative capability.

## 6 Summary and Assess

In the domain of CNN, feature fusion techniques are primarily manifested in two aspects: multi-scale feature fusion and cross-layer feature fusion. Multi-scale feature fusion involves the integration of feature information across different scales, allowing the model to simultaneously capture both global context and local details, thereby enhancing its performance in tasks such as image recognition and object detection. Cross-layer feature fusion, on the other hand, entails the blending of features from different network layers. This technique combines low-level features containing fine-grained detail with high-level features embodying abstract semantic information. By doing so, it reinforces the expressive power and robustness of model.

Secondly, in ViT models, feature fusion techniques also play a critical role. When tackling sequential data and image recognition tasks, ViT combines features from different Transformer layers to garner richer contextual information and more precise feature representations. This style of feature fusion not only enhances the robustness of model but also boosts its generalizability, enabling it to cope more effectively with complex and varied input data.

Furthermore, the integration of NAS technology with feature fusion techniques provides a fresh perspective for model optimization. NAS technology, through its automated search strategies, identifies optimal network architectures and feature fusion methodologies. During the feature fusion process, NAS adapts the fusion strategies and network parameters according to the requirements of the task and the characteristics of the data, thereby yielding models with improved performance. This combination not only enhances the efficiency of model design but also imbues the models with increased adaptability and generalization capabilities.

Finally, GCN, as specialized neural network models for processing graph-structured data, exhibit unique applications in the realm of feature fusion technology. In the context of GCN, the application of feature fusion techniques mainly manifests in two aspects: node feature aggregation and global information fusion. Node feature aggregation refers to the process where GCN gather and combine the attributes of neighboring nodes to update the representation of the target node, encapsulating local neighborhood characteristics. Global information fusion, on the other hand, signifies the holistic incorporation of information from all nodes in the graph, facilitating the transmission and aggregation of global context.

We have summarized the quantitative experimental results of some classic methods on the MS-COCO dataset [39], all of which use Faster RCNN as the basic model [40]. The evaluation indicators adopt single class average accuracy $AP_S$, $AP_M$, $AP_L$, $AP_{50}$, $AP_{75}$, where $S$ represents the category of smaller-sized objects in the dataset, $M$

represents the category of medium-sized objects, and $L$ represents the category of larger-sized objects, 50 and 70 represent $IoU$ thresholds. The Params and GFLOPs represent parameters and Giga Floating-point Operations Per Second , respectively. The results are shown in **Table 1**. In addition, we have summarized the mechanisms and access links of classic models combining feature fusion techniques with CNN, ViT, GCN, and NAS, as shown in **Table 2**.

**Table 1.** Quantitative detection results of some classic methods on the MS COCO dataset.

| Method | Size | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ | Params | GFLOPs |
|---|---|---|---|---|---|---|---|---|---|
| Auto-FPN [27] | 640×640 | 40.5 | 61.5 | 43.8 | 25.6 | 44.9 | 51.0 | 32.6 | - |
| NAS-FPN [28] | 640×640 | 37.7 | 54.5 | 41.1 | 5.5 | 44.5 | 56.9 | 68.2 | 103.0 |
| FPT [24] | 800×1000 | 38.0 | 57.1 | 38.9 | 20.5 | 38.1 | 55.7 | 88.2 | 346.2 |
| GraphFPN [33] | 800×1000 | 39.1 | 58.3 | 39.4 | 22.4 | 38.9 | 56.7 | 100.0 | 380.0 |
| LFPN [34] | 800×1000 | 38.7 | 60.4 | 41.9 | 23.6 | 42.5 | 49.2 | - | - |

**Table 2.** The mechanism and access links of classic models combining feature fusion techniques with CNN, ViT, GCN, and NAS.

| Type | Method | Mechanism and Access Links |
|---|---|---|
| CNN | ResNet[7] | Residual Connections, Deep Network Design<br>Access Links: https://github.com/GarsonWw/resnet-garson.git |
| | DenseNet[12] | Dense Connectivity, Feature Reuse<br>Access: Links: https://github.com/liuzhuang13/DenseNet |
| | HRNet[13] | High-Resolution Representations, Progressive Parallel Connections<br>Access: Links: https://github.com/HRNet |
| | CEDNet[14] | Spatial Feature Interaction Token Mixer, Channel Feature Interaction<br>Access: Links: https://github.com/zhanggang001/CEDNet |
| ViT | FPT[24] | Bidirectional Information Propagation Paths, Adaptive Feature<br>Access Links: https://github.com/dongzhang89/FPT |
| | PVT[25] | Parallel Feature Pyramids, Spatial Pyramid Pooling<br>Access Links: https://github.com/whai362/PVT |
| NAS | Auto-FPN[27] | Fully Connected Search Space, Optimal Search, Continuous Relaxation |
| | NAS-FPN[28] | Scalable Search Space, Modular Search, Merge Units<br>Access Links: https://github.com/open-mmlab/mmdetection |
| GCN | GraphFPN[33] | Part-Whole Hierarchy, Dynamic Modeling<br>Access Links: https://github.com/GangmingZhao/GraphFPN-Graph-Feature-Pyramid-Network-for-Object-Detection |
| | LFPN[34] | Latent Space, Intra-Scale Fusion, Cross-Scale Fusion |

## 7 Conclusion

The research on feature fusion technology has become the mainstream in the field of computer vision, and efficient and high-precision model structures have become the goal pursued by countless researchers. The feature fusion method, due to its powerful feature reuse ability, integrates multi-scale features and integrates contextual information during the fusion process, thereby extracting more discriminative features, improving the robustness and generalization ability of the algorithm, which is very beneficial for improving the accuracy of object detection.

In this article, we provide a detailed introduction to the combination of feature fusion technology with four technologies: CNN, ViT, GCN, and NAS. We list several classic models and explain the key role of feature fusion ideas in each model. Finally, we have conducted a comprehensive summary, hoping to provide a clear idea for subsequent related research.

**Acknowledgments.** This research was supported in part by the National Natural Science Foundation project (No.62066033), Inner Mongolia Natural Science Foundation Outstanding Youth Fund project (No.2022JQ05), Inner Mongolia Autonomous Region Science and Technology Plan project (No.2021GG0158), Hohhot City University-Institute Collaborative Innovation project, and Inner Mongolia University Young Scientific and Technological Talent Cultivation project (No.21221505)

## References

1. Hariharan B., Arbeláez P., Girshick R., Malik J.: Hypercolumns for object segmentation and fine-grained localization. In: CVPR, pp. 447-456 (2015)
2. Bell S., Zitnick C.L., Bala K., Girshick R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR, pp. 2874-2883 (2016)
3. Redmon J., Farhadi A.: YOLO9000: better, faster, stronger. In: CVPR, pp. 7263-7271 (2017)
4. He K., Zhang X., Ren S., Sun J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence. **37**(7), 1904-1916 (2015)
5. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich V.: Going deeper with convolutions. In: CVPR, pp. 1-9 (2015)
6. Quan Y., Zhang D., Zhang L., Tang J.: Centralized feature pyramid for object detection[J]. IEEE Transactions on Image Processing. **32**, 4341-4354 (2023)
7. He K., Zhang X., Ren S., Sun J.: Deep residual learning for image recognition. In: CVPR, pp. 770-778 (2016)
8. Liu Z., Mao H., Wu C.Y., Feichtenhofer C., Darrell F., Xie S.: A convnet for the 2020s. In: CVPR, pp. 11976-11986 (2022)
9. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly: S., Uszkoreit J., Houlsby N.: An image is worth 16x16 words: Transformers for image recognition at scale. Arxiv. **2010**.11929 (2020). https://doi.org/10.48550/arXiv.2010.11929

10. Kipf T.N., Welling M.: Semi-supervised classification with graph convolutional networks. Arxiv. **1609**.02907 (2016). https://doi.org/10.48550/arXiv.1609.02907
11. Liu C., Zoph B., Neumann M., Shlens J., Hua W., Li L.J., Fei L.F., Yuille A., Huang J., Murphy K.: Progressive neural architecture search. In: ECCV, pp. 19-34 (2018)
12. Huang G., Liu Z., Van Der Maaten L., Weinberger K.Q.: Densely connected convolutional networks. In: CVPR, pp. 4700-4708 (2017)
13. Wang J., Sun K., Cheng T., Jiang B., Deng C., Zhao Y., Liu D., Mu Y., Tan M., Wang X., Liu W., Xiao B.: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence. **43**(10), 3349-3364 (2020)
14. Zhang G., Li Z., Tang C., Li J., Hu X.: CEDNet: A Cascade Encoder-Decoder Network for Dense Prediction. Arxiv. **2302**.06052 (2023). https://doi.org/10.48550/arXiv.2302.06052
15. Krizhevsky A., Sutskever I., Hinton G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS, pp. 25 (2012)
16. Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. Arxiv. **1409**.1556 (2014). https://doi.org/10.48550/arXiv.1409.1556
17. LeCun Y., Bottou L., Bengio Y., Haffner P.: Gradient-based learning applied to document recognition. Science. **86**(11), 2278-2324 (1998)
18. Lin T.Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S.: Feature pyramid networks for object detection. In: CVPR, pp. 2117-2125 (2017)
19. Tan M., Pang R., Le Q.V.: Efficientdet: Scalable and efficient object detection.In: CVPR, pp. 10781-10790 (2020)
20. Liu Z., Lin Y., Cao Y., Hu H., Wei Y., Zhang Z., Lin S., Guo B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV, pp. 10012-10022 (2021)
21. Newell A., Yang K., Deng J.: Stacked hourglass networks for human pose estimation. In: ECCV, pp. 483-499 (2016)
22. Ronneberger O., Fischer P., Brox T.: U-net: Convolutional networks for biomedical image segmentation. In MICCAI, pp. 234-241 (2015)
23. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jonese L., Gomez A.N., Kaiser L., Polosukhin I.: Attention is all you need. In: NeurIPS, pp. **30** (2017)
24. Zhang D., Zhang H., Tang J., Wang M., Hua X., Sun Q.: Feature pyramid transformer. In: ECCV, pp. 323-339 (2020)
25. Wang W., Xie E., Li X., Fan D.P., Song K., Liang D., Lu T., Luo P., Shao L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV, pp. 568-578 (2021)
26. Wang X., Girshick R., Gupta A., He K.: Non-local neural networks. In: CVPR, pp. 7794-7803 (2018)
27. Xu H., Yao L., Zhang W., Liang X., Li Z.: Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In: ICCV, pp. 6649-6658 (2019)
28. Ghiasi G., Lin T.Y., Le Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: CVPR, pp. 7036-7045 (2019)
29. Real E., Moore S., Selle A., Saxena S., Suematsu Y.L., Tan J., Le Q.V., Kurakin A.: Large-scale evolution of image classifiers. In: ICML, pp. 2902-2911 (2017)
30. Zhong Z., Yan J., Wu W., Shao J., Liu C.L.: Practical block-wise neural network architecture generation. In: CVPR, pp. 2423-2432 (2018)
31. Xie S., Zheng H., Liu C., Lin L.: SNAS: stochastic neural architecture search. Arxiv. **1812**.09926 (2018). https://doi.org/10.48550/arXiv.1812.09926
32. Zaremba W., Sutskever I., Vinyals O.: Recurrent neural network regularization. Arxiv. **1409**.2329 (2014). https://doi.org/10.48550/arXiv.1409.2329

33. Zhao G., Ge W., Yu Y.: GraphFPN: Graph feature pyramid network for object detection. In: ICCV, pp. 2763-2772 (2021)
34. Xie J., Pang Y., Nie J., Cao J., Han J.: Latent feature pyramid network for object detection. IEEE Transactions on Multimedia. **25**, 2153-2163 (2022)
35. Qiao S., Chen L.C., Yuille A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: CVPR, pp. 10213-10224 (2021)
36. Hu J., Shen L., Sun G.: Squeeze-and-excitation networks. In: CVPR, pp. 7132-7141 (2018)
37. Fu J., Liu J., Tian H., Li Y., Bao Y., Fang Z., Lu H.: Dual attention network for scene segmentation.In: CVPR, pp. 3146-3154 (2019)
38. Chen Y., Rohrbach M., Yan Z., Yan S., Feng J., Kalantidis Y.: Graph-based global reasoning networks. In: CVPR, pp. 433-442 (2019)
39. Lin T.Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollar P., Zitnick C.L.: Microsoft coco: Common objects in context. In: ECCV, pp. 740-755 (2014)
40. Ren S., He K., Girshick R., Sun J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS, pp. **28** (2015)