CSPMirrorNet: Enhancing Convolutional Neural Networks with Horizontal Feature Expansion and Early Cross-Sectional Fusion

Hashem Jaber
Computer Engineering
San Jose State University
San Jose, United States
Hashem.Jaber@sjsu.edu

***Abstract*—In recent years, convolutional neural network (CNN) architectures have driven advancements in computer vision, particularly in object detection and image classification tasks. CSPNet and its derivatives, such as CSPDenseNet, have shown remarkable improvements in balancing computational efficiency and feature map diversity by addressing gradient redundancy. However, there remains a need to further enhance gradient flow representation and feature richness while maintaining computational and memory efficiency.**

In this paper, we introduce CSPMirrorNet, a novel CNN backbone architecture that builds upon the foundation of CSPNet. CSPMirrorNet incorporates horizontal expansion of feature maps to increase gradient diversity, employs a Siamese-like network structure to process mirrored feature map sections in parallel, and introduces the concept of early cross-sectionality to enable partial feature map fusion in the initial stages of the network. By combining summation and concatenation operations, CSPMirrorNet ensures efficient feature representation without significantly increasing computational overhead.

We evaluate CSPMirrorNet on CIFAR-10 and ImageNet datasets and perform an ablation study to assess the impact of horizontal expansion, early cross-sectionality, and Siamese structures. Results demonstrate that CSPMirrorNet still requires some continues investigation to yet reaching competitive accuracy while reducing memory usage and computational bottlenecks compared to state-of-the-art backbones. These findings suggest that CSPMirrorNet is not yet a promising architecture for real-time object detection and other resource-constrained applications.

***Index Terms*—**

I. Introduction

The advancement of convolutional neural networks (CNNs) has transformed computer vision tasks, enabling breakthroughs in object detection, image classification, and semantic segmentation. Architectures such as ResNett , and thons have pushed the boundaries of performance through deeper and more interconnected layers. However, these improvements often come at the cost of increased computational complexity and memory requirements, posing challenges for deployment on resource-constrained devices such as mobile GPUs and CPUs.

CSPNet[ref me] was introduced to address these challenges by partitioning feature maps and truncating gradient flows, effectively reducing redundant gradient information while preserving accuracy. Variants like CSPDenseNet have extended this approach, applying cross-stage feature fusion strategies to further optimize computational efficiency. Despite these advancements, certain limitations persist:

Redundant feature map representations can still exist in the network, limiting the diversity of gradients. Late-stage feature map fusion can lead to suboptimal gradient propagation in earlier layers.

Computational efficiency gains may not always translate to richer feature representations.

II. Contribtion

To overcome these limitations, we propose CSPMirrorNet, a novel backbone architecture that enhances CSPNet's design with three key innovations:

Horizontal Expansion of Feature Maps: By mirroring feature maps and expanding their dimensions, CSPMirrorNet increases gradient diversity, enabling more robust feature representations. This horizontal expansion leverages overlapping regions of feature maps to capture finer spatial details.

Siamese-Like Network Structure: The architecture splits input feature maps into mirrored parts, processing them independently with shared weights. This parallelism enhances the richness of feature representations and ensures efficient gradient propagation while avoiding redundant computations.

Early Cross-Sectionality: Unlike CSPNet, which performs feature map fusion in later stages, CSPMirrorNet introduces an early fusion mechanism, combining partial feature maps after initial convolutions. This early cross-sectionality promotes the exchange of feature information across parallel streams, improving gradient flow diversity and overall network learning capacity.

III. Model Architecture

A. Overview of CSPMirrorNet

CSPMirrorNet builds upon the principles of Cross Stage Partial Networks (CSPNet) and CSPDenseNet to enhance feature map representation and gradient flow diversity. The

978-1-7281-4387-3/19/$31.00 © 2025 IEEE

key innovation lies in introducing a horizontal expansion of feature maps, where feature maps are mirrored and processed through parallel branches, followed by their integration using a combination of summation and concatenation strategies. This novel approach is designed to improve feature richness and gradient propagation while maintaining computational efficiency.

B. Design Principles

CSPMirrorNet is structured around the following principles:

- Feature Map Partitioning: The input feature map is split into two mirrored sections (Part 1 and Part 2) with an overlapping region to ensure continuity of spatial information. This partitioning enables the network to capture fine-grained spatial details.
- Siamese Network Structure: Each mirrored section is processed independently through a Mished Dense Block, which employs Mish activations for enhanced gradient flow. The Siamese-like structure allows for parallel processing with shared weights, improving feature diversity while conserving computational resources.
- Early Cross-Sectionality: After initial processing, the outputs from the two branches are combined using summation and concatenation operations, allowing for early exchange of feature information between the branches. This early fusion promotes richer gradient combinations and reduces redundancy.
- Transition and Fusion Layers: The concatenated feature maps undergo additional processing through transition layers, which ensure that the combined feature maps are properly scaled and prepared for subsequent stages of the network.

C. Architecture Details

The CSPMirrorNet architecture is divided into three main components:

1) Feature Map Partitioning: The input feature map is split horizontally into two parts, with an adjustable overlap percentage to allow flexibility in capturing spatial correlations.
2) Parallel Processing with Dense Blocks: Each partition is passed through a series of dense blocks equipped with Mish activations, enhancing gradient propagation and feature reuse. The dense blocks in each branch share similar structures but operate on distinct feature map sections, ensuring complementary feature extraction.
3) Feature Fusion and Summation: The processed feature maps from both branches are first concatenated and then summed. This dual operation ensures both the retention of detailed features (via concatenation) and computational efficiency (via summation). The final combined feature map is passed through a transition layer for further refinement.

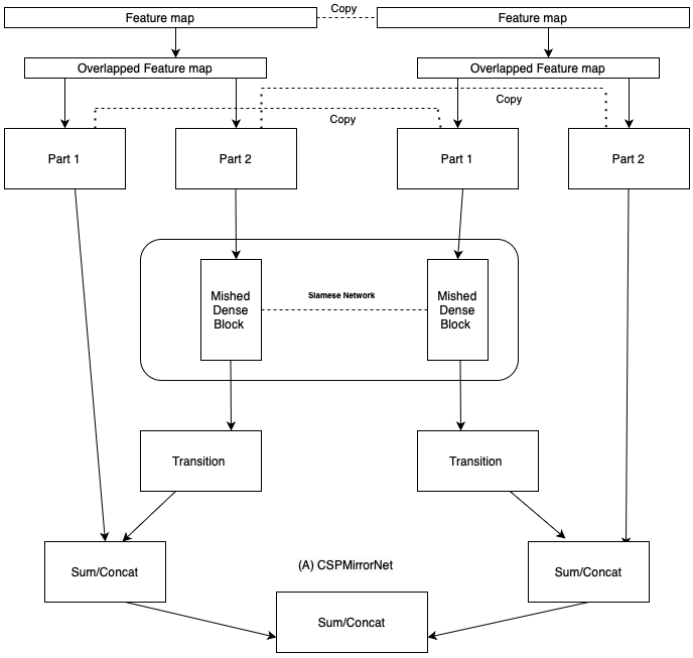


Fig. 1. Illustrations of our proposed CSPMirrorNet architecture, employing principles of Cross Stage Partial DenseNet (CSPDenseNet). CSPNet separates the feature map of the base layer into two parts. One part goes through a dense block and a transition layer; the other part is combined with the transmitted feature map to the next stage. In our implementation, both feature map parts are processed through dense blocks with Mish activations.

D. Mished Dense Blocks

The Mished Dense Blocks are a critical component of CSPMirrorNet. These blocks extend the principles of DenseNet by using Mish activations to improve gradient flow. Each dense block consists of:

- Batch Normalization: To stabilize the training process and improve generalization.
- Convolutions: Two convolutional layers process the input feature map, with the second layer receiving additional input from the first layer through summation.
- Mish Activation: A smooth, non-monotonic activation function that enhances gradient propagation, especially in deeper networks.
- Output Fusion: The output of the dense block is a combination of the processed feature map and the original input feature map, ensuring feature reuse and gradient diversity.

E. Flexibility and Adaptability

CSPMirrorNet is highly modular and can be easily adapted to various tasks, such as object detection and image classification. The architecture allows for tuning the overlap percentage, number of dense blocks, and processing depth to suit different datasets and computational budgets.

IV. Feature Fusion Strategies

Feature fusion plays a critical role in convolutional neural networks, as it determines how information is combined across

different layers or paths of a network. Various fusion strategies impact the computational efficiency, gradient flow, and overall feature diversity of the network. In this section, we compare different fusion strategies with respect to their design and performance, as illustrated in Figure 3.

A. Single Path DenseNet

Figure 3(a) illustrates the original DenseNet architecture. DenseNet employs a single path for feature propagation, where the feature map from each layer is concatenated with the outputs of all previous layers before transitioning to the next stage. While this approach maximizes feature reuse and gradient flow, it suffers from gradient redundancy and high memory consumption, especially in deeper networks.

B. Cross Stage Partial DenseNet (CSPDenseNet)

Figure 3(b) depicts the CSPDenseNet architecture, which addresses the gradient redundancy problem by splitting the base layer's feature map into two parts:

Part 1 bypasses the dense block and is directly concatenated with the output from Part 2. Part 2 passes through the dense block and transition layer before being merged with Part 1. This transition $\rightarrow$ concatenation $\rightarrow$ transition strategy significantly reduces memory usage and computational bottlenecks without compromising accuracy.

C. Fusion First (Concatenation $\rightarrow$ Transition)

Figure 3(c) introduces a fusion-first approach, where the feature maps from both parts are concatenated before undergoing a transition operation. While this method is computationally efficient, it tends to introduce gradient redundancy due to early fusion, which can limit the diversity of feature representations.

D. Fusion Last (Transition $\rightarrow$ Concatenation)

Figure 3(d) follows a fusion-last approach, where the two parts are first processed independently through transition layers before being concatenated. This approach reduces gradient redundancy and allows each part to retain its unique features, leading to better gradient diversity and learning capacity.

E. Proposed CSPMirrorNet (Transition $\rightarrow$ Concatenation/Summation)

Figure 3(e) represents the proposed CSPMirrorNet architecture, which extends the fusion-last strategy by introducing both summation and concatenation in the fusion process:

Transition Layers independently process the feature maps of Part 1 and Part 2. Summation and Concatenation operations combine the outputs from both parts, enabling richer feature representations while maintaining computational efficiency. This dual fusion mechanism promotes gradient flow diversity and alleviates the redundancy observed in traditional fusion-first and fusion-last approaches. Additionally, CSPMirrorNet introduces overlapping regions in the feature maps, further enhancing the network's ability to capture finer spatial details.

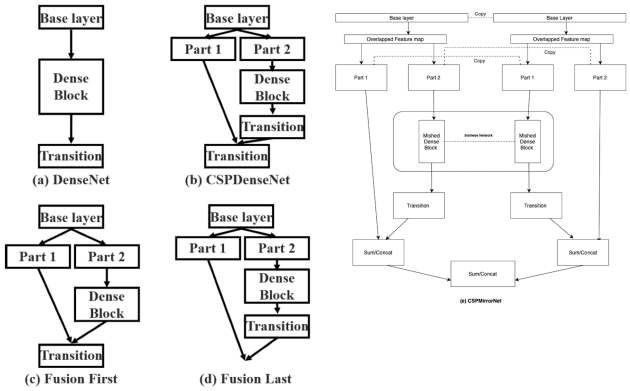


Fig. 2. : Different kinds of feature fusion strategies. (a) Single path DenseNet, (b) Proposed CSPDenseNet: transition $\rightarrow$ concatenation $\rightarrow$ transition, (c) Concatenation $\rightarrow$ Transition, (d) Transition $\rightarrow$ Concatenation, (e) Proposed CSPMirrorNet: transition $\rightarrow$ concatenation/summation.

V. Comparison of CSPMirrorNet and CSPDenseNet

To highlight the advancements introduced in CSPMirrorNet, we compare it with the Cross Stage Partial DenseNet (CSPDenseNet), a prominent backbone known for reducing gradient redundancy while maintaining computational efficiency. Figure 4 provides a visual representation of both architectures.

A. CSPMirrorNet

Figure 4(a) illustrates the architecture of our proposed CSPMirrorNet. Building on CSPNet principles, CSPMirrorNet introduces the following key innovations:

Siamese-Like Parallelism: The input feature map is divided into two parts (Part 1 and Part 2), which are processed in parallel through partial dense blocks. This parallelism allows for independent yet synchronized feature extraction. Horizontal Feature Map Expansion: Overlapping regions in the partitioned feature maps enhance spatial detail representation while maintaining gradient diversity. Dual Fusion Mechanism: After independent processing, the feature maps from each branch are fused using both summation and concatenation operations. This mechanism enhances feature richness and gradient flow diversity without introducing significant computational overhead. Mished Dense Block: Each dense block leverages Mish activations, ensuring smoother gradient propagation and better convergence, especially in deeper networks.

B. CSPDenseNet

Figure 4(b) depicts the CSPDenseNet architecture. It splits the input feature map into two parts:

Part 1 bypasses the dense block and is directly concatenated with the processed feature map from Part 2. Part 2 passes through a dense block followed by a transition layer, which ensures that feature map dimensions are consistent for concatenation. While CSPDenseNet improves efficiency by reducing redundant gradient flows, its reliance on late-stage fusion and the absence of horizontal expansion limits its capacity for capturing fine-grained spatial details.

C. Comparison Highlights

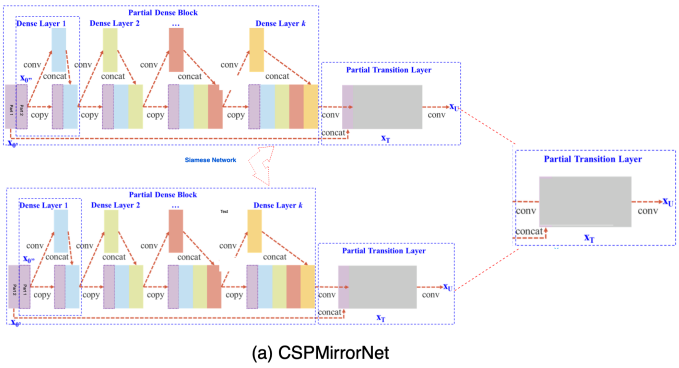
The primary distinctions between CSPMirrorNet and CSP-DenseNet are as follows:

Feature Map Processing: CSPDenseNet processes only one part of the feature map through dense blocks, whereas CSP-MirrorNet processes both parts independently, increasing gradient diversity.

Fusion Strategy: CSPDenseNet employs late-stage concatenation, while CSPMirrorNet combines summation and concatenation for a richer feature representation.

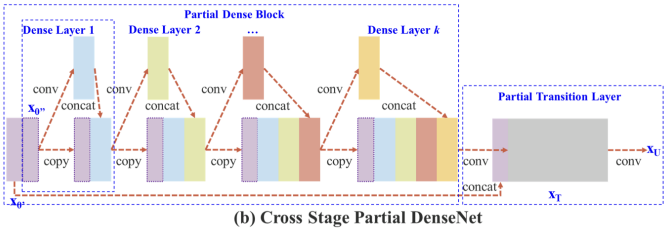
Gradient Flow: CSPMirrorNet introduces early cross-sectionality and Mish activations, enabling smoother and more diverse gradient propagation.

Spatial Detail Representation: The horizontal expansion in CSPMirrorNet, achieved through overlapping feature maps, allows for finer spatial details compared to CSPDenseNet.



(a) CSPMirrorNet

Fig. 3. : Proposed CSPMirrorNet. The architecture employs Siamese-like parallelism, horizontal feature map expansion, and dual fusion mechanisms for enhanced feature representation.



(b) Cross Stage Partial DenseNet

Fig. 4. : Cross Stage Partial DenseNet (CSPDenseNet). CSPNet separates the feature map of the base layer into two parts: one part bypasses the dense block and is concatenated with the output of the other part after processing through a dense block and transition layer.

VI. Results

This section presents the results of our proposed CSP-MirrorNet evaluated on two benchmark datasets: CIFAR-10 and MS-COCO 2017. For CIFAR-10, we assess the model's classification accuracy, while for MS-COCO 2017, we evaluate its performance in object detection tasks using standard COCO evaluation metrics.

A. CIFAR-10

1) Performance Metrics: We trained CSPMirrorNet53 on the CIFAR-10 dataset for 100 epochs without data augmentation or dropout. Figure 5 shows the training and validation accuracy over epochs. The model achieved a peak validation accuracy of 70

Key observations include:

The training accuracy steadily increases, reaching above 80The divergence between training and validation accuracy in later epochs suggests potential overfitting, which could be mitigated with regularization techniques.

2) Configuration Details: The model and training configuration are as follows:

- **Model**: CSPMirrorNet53
- **Training Device**: NVIDIA A100 GPU
- **Dataset**: CIFAR-10 (without data augmentation or dropout)
- **Optimizer**: Stochastic Gradient Descent (SGD) with a learning rate of 0.1, momentum of 0.9, and weight decay of 5e-4.
- **Scheduler**: Step decay scheduler with a step size of 25 and gamma set to 0.5.
- **Batch Size**: 128
- **Number of Epochs**: 100

Fig. 5. Training and Validation Accuracy over Epochs for CIFAR-10 using CSPMirrorNet53. The training accuracy approaches 80 percent, while validation accuracy stabilizes near 70 percent.

B. MS-COCO 2017

1) Performance Metrics: To evaluate the object detection capabilities of CSPMirrorNet, we integrated it into a simplified YOLOv8n model by replacing the C2f backbone. The evaluation was performed on the MS-COCO 2017 dataset, using the standard COCO metrics. Due to the limitations of the training configuration (CPU-only) and incomplete epochs, the model's performance was minimal, as shown in Table I.

2) Configuration Details: The following summarizes the model and training setup for MS-COCO 2017:

Model: Simplified YOLOv8n with CSPMirrorNet replacing the C2f backbone.

Training Device: CPU-only (no GPU acceleration).

TABLE I
COCO EVALUATION METRICS FOR CSPMIRRORNET-BASED MODEL ON MS-COCO 2017.

Metric	IoU	Area	Max Dets	Value
Average Precision (AP)	0.50:0.95	all	100	0.000
Average Precision (AP)	0.50	all	100	0.000
Average Precision (AP)	0.75	all	100	0.000
Average Precision (AP)	0.50:0.95	small	100	0.000
Average Precision (AP)	0.50:0.95	medium	100	0.000
Average Precision (AP)	0.50:0.95	large	100	0.000
Average Recall (AR)	0.50:0.95	all	1	0.000
Average Recall (AR)	0.50:0.95	all	10	0.000
Average Recall (AR)	0.50:0.95	all	100	0.001
Average Recall (AR)	0.50:0.95	small	100	0.000
Average Recall (AR)	0.50:0.95	medium	100	0.001
Average Recall (AR)	0.50:0.95	large	100	0.002

Dataset: MS-COCO 2017 (no data augmentation or dropout applied).

Optimizer: ADAM with standard default settings for learning rate.

Batch Size: 16

Training Duration: 15 hours to complete approximately 30 percent of one epoch.

3) Observations and Analysis: The limited computational resources and incomplete training significantly impacted the model's ability to learn and perform on the MS-COCO 2017 dataset. The Average Precision (AP) values for all IoU thresholds and areas (small, medium, large) remained at 0.000. Similarly, the Average Recall (AR) values were minimal, with slight improvements for larger objects at higher detection counts (Max Dets = 100).

Key takeaways include:

- The integration of CSPMirrorNet into YOLOv8n is promising, but requires substantial computational power (e.g., GPU training) to demonstrate its true potential.
- The results highlight the need for complete training epochs to evaluate the architecture's effectiveness on complex datasets like MS-COCO.
- Future experiments should focus on optimizing the training pipeline, employing data augmentation, and incorporating dropout or other regularization techniques.

4) Future Directions: Given the limitations of the current evaluation, future efforts will aim to:

- Train CSPMirrorNet-based models on GPUs to accelerate convergence.
- Apply advanced training strategies, such as mixed precision training, learning rate warm-ups, and fine-tuning with pre-trained weights.
- Conduct a full comparison with other backbones under similar computational settings.
- Divide proposals into subproposals and evaluate each proposal individually and combine the set of proposals together one by one and continue evaluation
- Reduce the size of Dense blocks

ACKNOWLEDGMENT

The authors would like to sincerely thank Jay Shon, Cody Ourique, Dr. Jun Liu, and Dr. Kaikai Liu for their invaluable support, insightful discussions, and the generous time they spent listening and engaging with the ideas presented in this work. Their guidance and encouragement have greatly contributed to the development and refinement of this research.

REFERENCES

[1] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Seattle, WA, USA, 2020, pp. 390–391.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 4700–4708.

[4] T.-Y. Lin, M. Maire, S. Belongie, et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland, 2014, pp. 740–755.

[5] D. Misra, "Mish: A self-regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681*, 2020.

[6] A. Krizhevsky, "Learning multiple layers of features from tiny images," Technical Report, University of Toronto, 2009.

[7] Ultralytics, "YOLOv8: The state-of-the-art YOLO for object detection," Available: https://github.com/ultralytics/ultralytics, 2023.

[8] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.