Dean&Francis
ISSN 2959-6157

Exploring Backbone Network Choices in FCOS3D: Performance and Efficiency Analysis

Linyi He

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798, Singapore
E-mail: HE0005YI@e.ntu.deu.sg

Abstract:

This study presents a comparative analysis of the performance of two modified object detection models, FCOS-Swin and FCOS-ConvNeXt, against the FCOS3D baseline model using the nuScenes dataset. The study evaluates the models based on their classification results for various categories of objects and on multiple evaluation metrics. We compare FCOS-Swin and FCOS-ConvNeXt, which utilize different backbone architectures, to evaluate their effectiveness in 3D object detection. Results show that the modified models exhibit comparable performance with slight variations in all metrics compared to the baseline, but fall short of the fine-tuned FCOS3D model. Potential reasons for this performance gap, including model parameter size, data augmentation methods, learning rate settings, and training epochs, are discussed. This study also explores possible improvements and future work, such as switching to larger backbone models, utilizing stronger data augmentation techniques, adjusting the learning rate method, increasing training epochs, and incorporating temporal and spatial logic to optimize model performance.

Keywords: 3D object detection, FCOS3D, FCOS-Swin, FCOS-ConvNeXt, nuScenes dataset, performance comparison

1. Introduction

Object detection is an important application of computer vision, playing a vital role in industrial applications, particularly in the domain of autonomous driving [1]. Autonomous car need to precisely identify various road objects, including vehicles, pedestrians, and traffic signs, to perform decision-making and control. In 3D object detection, multi-sensor fusion approaches, such as the integration of multiple cameras or the combination of LiDAR, RaDAR and cameras, have become main solutions currently due to their ability to achieve high detection accuracy and robustness. Additionally, methods like Bird's Eye View (BEV) [2] or occupancy prediction [3] have also conducted impressive performance. However, within these frameworks, an efficient backbone network is crucial for feature extraction, especially in

ISSN 2959-6157

purely vision-based solutions.

Given the crucial role of efficient backbone networks, this study investigates the impact of backbone network substitution on the performance of FCOS3D [4], a powerful monocular 3D object detection framework. This study focuses investigating the impact of backbone network substitution on 3D object detection performance. FCOS [5], a fully convolutional, one-stage, anchor-free 2D object detection algorithm, has gained widespread usage for its pixel-by-pixel prediction and streamlined training process. FCOS3D, built on FCOS [5], modified the approach for 3D object detection by predicting 3D parameters, enabling monocular 3D object detection and show excellent performance on datasets like nuScenes [6].

The contributions of this study are two below:

· Validation of Backbone Replacement Effectiveness: By constructing FCOS3D-Swin and FCOS3D-ConvNeXt networks, this study compares the performance of replacing the original ResNet101 [7] backbone with Swin Transformer [8] and ConvNeXt [9] in 3D object detection tasks. Experimental results indicate that the updated backbone networks achieve comparable detection performance with slight variations in mAP and NDS to the original network while utilizing fewer parameters.

· Demonstration of Backbone Network Effectiveness: In contrast to complex state-of-the-art (SOTA) solutions, this study presents a straightforward approach to enhancing 3D object detection performance by solely improving the backbone network. This method reduces system complexity and enhances computational efficiency, offering a acceptable perspective for purely vision-based 3D object detection.

Although the proposed improvement strategy achieves results comparable to the original network, it still falls behind the current SOTA purely vision-based solutions. This disparity primarily stems from the exclusion of complex techniques such as multi-sensor fusion and temporal information utilization. Nevertheless, this research provides valuable insights for purely vision-based 3D object detection, holding both theoretical and practical significance.

2. Related Work

2.1 2D Object Detection

Numerous methods exist for 2D object detection. Based on whether the detector generates anchors first, there are two kinds of main architectures: anchor-based and anchor-free. Anchor-based architectures, such as R-CNN [9] and its improved versions detect objects by regressing object bounding boxes directly from image features. In contrast, anchor-free architectures include the YOLO series [11], which perform object detection using preset anchors. Furthermore, with the introduction of the proposal based algorithms, like FCOS [5], have become an effective direction. These methods first learn image features to generate object boxes and then select the optimal box using techniques like Non-Maximum Suppression (NMS).

The introduction of the Transformer architecture and the self-attention mechanism has revolutionized the image domain. Vision Transformer (ViT) [12] achieves object recognition by dividing images into pixel patches, treating them as tokens, and performing position encoding and feature learning. Based on this, the DETR network was developed, which combines CNN feature extraction with Transformer processing for efficient object detection. To address issues related to image size limitations and detection speed in DETR , improvements such as deformable DETR [13] and RT-DETR [14] have been proposed. Additionally, the Swin Transformer [8], which adopts a windowing approach inspired by CNNs, optimizes the computational efficiency of Transformers for larger images. Subsequent architectures, such as ConvNeXt [9], which is comparable to Swin, continue to be explored and optimized.

2.2 3D Object Detection

3D object detection is more complex due to the incorporation of depth information, which is not preserved in regular images. Based on how depth information is obtained, 3D object detection can be classified into point cloud-based methods, pure vision-based methods, and hybrid methods.

Point cloud-based methods such as PointRCNN [15] detects object by using raw cloud data, but their effectiveness is limited by the sparsity of point clouds. Therefore, techniques such as point cloud enhancement and voxelization, like PointPillar [16] and SECOND [17], have been introduced.

Pure vision-based methods are divided into monocular and multi-view vision. Monocular vision methods, such as FCOS3D [4], achieve 3D object detection by modifying the loss function of 2D networks. Multi-view vision methods, on the other hand, utilize information fusion from dual or multiple cameras to enhance depth perception, such as DSGN [18] and BEVFormer [19].

Among pure vision-based methods, the FCOS3D network is a significant monocular 3D object detection solution. It is based on a 2D backbone network and performs object detection by modifying the centerness loss function and adding loss functions that include depth, velocity, rotation, and other information. This method does not rely on correlation information between multiple cameras or between

consecutive video frames, making it a direct replacement for some existing 3D detection networks. Additionally, there are other monocular 3D detection methods such as CenterNet [20] and SMOKE [21], as well as works like PGD [22], which is based on FCOS3D and incorporates depth information estimation.

Multi-view vision methods further utilize information from multiple cameras. Dual-camera schemes, such as DSGN [18] and Stereo R-CNN [23], enhance depth perception through feature fusion. Multi-camera schemes, such as BEVFormer [19], utilize BEV feature processing and the Transformer architecture to project fused information to a Bird's Eye View (BEV) for object detection. The BEV principle involves three key steps: lifting 2D image features to 3D space, splatting them onto a bird's-eye-view representation, and shooting rays to detect objects [24]. Classic schemes that employ the BEV principle include BEVDet [25]. Among multi-view 3D object detection methods using the Transformer architecture, DETR3D [26] is also noteworthy.

Fusion methods combine information from multiple sensors for more accurate detection. Depending on the stage of fusion, they can be classified into early input fusion, mid-level feature fusion, and late result fusion. Input fusion methods, such as PointPainting [27], enhance point cloud information with images. Feature fusion methods, such as EPNet [28], improves detection accuracy by fusing features from different sensors. Result fusion methods, such as CLOCs [29], also achieves accurate detection by fusing detection results from different sensors.

3. Approach

3D object detection aims to accurately locate targets and provide key information of the targets in 3D space. This study focuses on exploring the impact on FCOS3D's [4] performance through replacing the backbone networks. This section is about three main aspects: firstly, a brief introduction to FCOS3D; secondly, an overview of Swin Transformer [8], the rationale for selecting, and modification to the original ResNet101 [7]; and thirdly, a repeat of above to ConvNeXt [9].

3.1 Overview of FCOS3D Architecture

The FCOS3D [4] network comprises three core components: a backbone network, a neck network, and multiple detection heads in Figure 1, which shows the comparison of the overall architectures of FCOS3D original with the modified FCOS3D-Swin and FCOS3D-ConvNeXt.

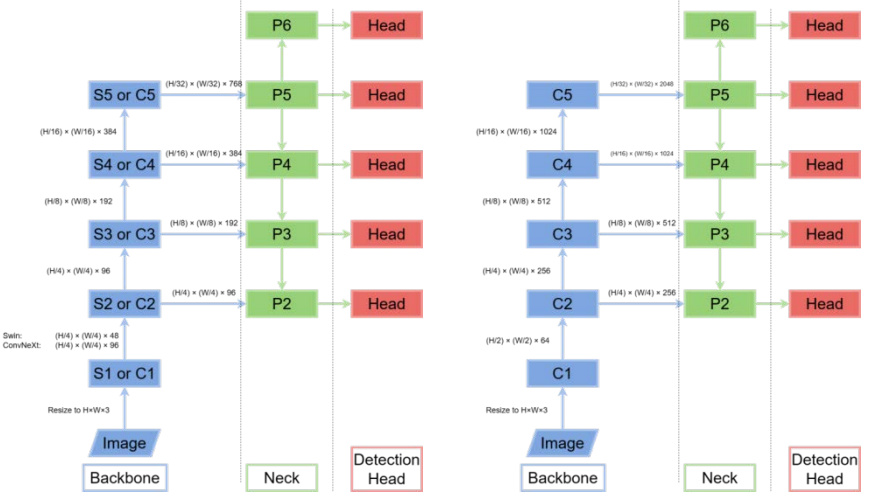

Figure 1 Comparison of Architecture of FCOS3D and modified networks in this study. The main adjustment to the original network is the adjustment of the input to the FPN Neck channel number.

3.1.1 Backbone Network

The backbone employs ResNet101 [7] with Deformable Convolutional Networks (DCN) [4] to extract image features effectively.

3.1.2 Neck Network

A Feature Pyramid Network (FPN) [4] serves as the neck, generating multi-scale feature maps (P2-P6) by fusing features from different CNN layers.

3.1.3 Detection Heads

Multiple heads perform various tasks:

· Object Classification: Predicts the object class probability (p) using a head similar to FCOS.

· 3D Centerness [4]: Calculates the 2D Gaussian distribution of the squared distance between the projected object

ISSN 2959-6157

center and the predicted bounding box center.

· Other Predictions: Estimates offsets ($\Delta x, \Delta y$), depth (d), size (w, l, h), rotation angle (θ), movement direction (C_θ), and velocity (v_x, v_y).

Loss function includes:

· Classification Loss:

$$L_{\text{cls}} = -\alpha(1-p)^{\gamma} \, logp$$

· Centerness Loss:

$$c = e^{-\alpha\left((\Delta x)^2+(\Delta y)^2\right)}$$

· Other Predictions Loss:

$$L_{\text{loc}} = \sum_{b\in\{\Delta x,\Delta y,d,w,l,h,\theta,v_x,v_y\}} \text{SmoothL1}(\Delta b)$$

· Total Loss:

$$L = \frac{1}{N_{\text{pos}}}\left(\beta_{\text{cls}}L_{\text{cls}} + \beta_{\text{attr}}L_{\text{attr}} + \beta_{\text{loc}}L_{\text{loc}} + \beta_{\text{dir}}L_{\text{dir}} + \beta_{\text{ct}}L_{\text{ct}}\right)$$

3.2 FCOS3D-Swin: Integration of Swin Transformer into the FCOS3D Framework

This section provides a brief introduction to the Swin Transformer [8], the rationale behind selecting, and details of the parameter modification for integration.

3.2.1 Introduction of Swin Transformer

The Swin Transformer [8] is an architecture modified from the Vision Transformer [12]. Compared to CNNs, the ViT [12] uses a global attention mechanism to enable global information interaction, but it suffers from high computational complexity. To address these issues, the Swin Transformer [8] introduces the following improvements:

· Hierarchical Feature Maps [8]: Enhanced feature perception of objects at different scales through downsamplin (e.g., 4x, 8x, 16x rates).

· Windows Multi-Head Self-Attention (W-MHSA) [8]: Reduced computational load by dividing the image into multiple windows and computing self-attention within each window.

· Shifted Windows Multi-Head Self-Attention (SW-MHSA) [8]: Enabled information exchange between different windows by shifting the windows, effectively fusing information across the entire image.

3.2.2 Rationale for Selecting Swin Transformer as the Backbone

The integration of the Swin Transformer [8] into FCOS3D [4] involved two main modifications:

· Superior 2D detection performance: The Swin Transformer outperformed many architectures in 2D object detection, replacing the ResNet [7] with the Swin Transformer is expected to yield a considerable 3D detection performance.

· Multi-scale feature learning capability: The Swin Transformer has its stage design similar to ResNet , enabling feature extraction of different sizes.

· Effective information exchange: Through W-MHSA and SW-MHSA, the Swin Transformer successfully enables communication between pixel blocks and surrounding features, enhancing the effectiveness of feature extraction.

3.2.3 Integration Procedure and Modifications of Swin Transformer

The integration of the Swin Transformer [8] into FCOS3D involved two main modifications:

· Network selection: The Swin Transformer tiny was chosen as the alternative backbone network to balance model performance and training time, due to its efficient resource utilization and strong 2D detection performance.

· Neck input adjustment: Similar to the role of ResNet in FCOS3D, the features output at each stage of the Swin Transformer need to be used as input for the neck network. Therefore, modifications in data channel number should be made to Swin Transformer.

3.3 FCOS3D-ConvNeXt: Integration of ConvNeXt into FCOS3D

This section provides a brief introduction to the ConvNeXt [9], the rationale behind selecting, and the adjustments for integration.

3.3.1 Introduction of ConvNeXt

ConvNeXt [9] is a network modified based on various existing architectures, aiming to prove that CNNs can still compete with Transformer-based architectures. Its key techniques and include:

· Macro Design [9]: Adjusting the stacking block ratio of stages, mimicking the structure of the Swin to 3:3:9:3, and improving the downsampling module by adopting a non-overlapping 4x4 convolution kernel with a stride of 4 for “patchify” operations.

· Convolution Change [9]: Introducing depthwise convolution to reduce computational load and enhance performance, which is similar to self-attention computation.

· Inverted Bottleneck [9]: Mimicking the MLP operation of Transformers, put dimension transformation before feature extraction and dimension scaling.

· Kernel Size [9]: Changing the mainstream 3x3 convolution kernel to 7x7 to imitate the Swin Transformer.

· Micro Design [9]: Replacing the ReLU activation function with GELU, reducing the use of activation functions, substituting Batch Normalization with Layer Normalization, and decreasing the use of normalization layers. These

adjustments have led to minor performance improvements.

3.3.2 Rationale for ConvNeXt as Backbone Selection

The selection of ConvNeXt [9] as the backbone network for FCOS3D [4] is mainly based on the following considerations:

· Outstanding Performance: ConvNeXt, through mimicking the Swin Transformer and absorbing the advantages of other architectures, has performed excellently in the computer vision domain .

· High Compatibility with FCOS3D: As both ConvNeXt and FCOS3D are convolutional neural networks, using ConvNeXt as the backbone is theoretically to encounter fewer problems, and the reuse of FCOS3D [4] related settings such as learning rates will be more reasonable.

· Comparability with Swin Transformer: ConvNeXt is designed to compete with the Swin Transformer. Using these two architectures in a study not only reduce code editing but also provides a better reference for performance comparisons.

3.3.3 Integration Process and Adjustments

The process of integrating ConvNeXt [9] into FCOS3D [4] is similar to that of the Swin Transformer [8], with minor adjustments to accommodate the specific architecture of ConvNeXt tiny.

4. Experimental Setup

This study uses the nuScenes v1.0 [6] dataset for model training, validation, and testing. All modifications and configurations of the models are conducted within the MMDetection3D framework utilized by FCOS3D [4]. The following sections provide a brief overview of the nuScenes [6] dataset, model architecture and configurations.

4.1 Dataset Introduction of Nuscenes

NuScenes v1.0 [6] involves 1,000 driving segments captured in various climates and times across four global cities. Each clip includes images from six camera perspectives, data from five differently oriented radars, and one surround-view LiDAR. The dataset contains over 1.4 million 3D detection boxes, covering ten categories of objects.

4.2 Model Specifications

This study modifies the FCOS3D framework by replacing the original ResNet101 backbone with either Swin Transformer tiny or ConvNeXt tiny. The neck network follows the FPN architecture used in FCOS3D [4] for feature extraction, processing, and fusion. The detection head adopts the structure designed by FCOS3D [4].

In terms of model configuration, to minimize the impact of configuration differences, this study reuses the original training configuration of FCOS3D [4]. The SGD optimizer is chosen following the original FCOS3D training configuration for consistency, which means AdamW which is widly used for training Transformer is not chosen. The SGD optimizer is set with an initial learning rate of 0.002, which decreases in a stepped manner to 0.0002 and 0.00002 as the number of training epochs increases. Experiments were conducted on an Ubuntu 20.04 system with a single NVIDIA 4090D GPU. The Python version was 3.8, the PyTorch version was 1.11.0 and the CUDA version was 11.3. Details of the specific setups should be referred to MMDetection3D [4].The batch size was set to 2, and training was performed for approximately 72 hours. Due to code modifications, distributed training was not conducted.

5. Results and Discussion

5.1 Comparison of Detection Results

Table 1 shows the detection performance of the proposed models (FCOS3D-Swin and FCOS3D-ConvNeXt) and the baseline FCOS3D [4] model on the nuScenes validation set.

Table 1 Comparison of Performance on NuScenes Validation Set.

CV = constructing vehicle, ped = pedestrians, TC = traffic cone.

Methods	car	truck	bus	trailer	CV	ped	motor	bicycle	TC	barrier	mAP
FCOS3D [4]	0.524	0.27	0.277	0.255	0.117	0.397	0.345	0.298	0.557	0.538	0.358
FCOS3D-Swin (Ours)	0.405	0.145	0.193	0.034	0.025	0.35	0.201	0.165	0.44	0.341	0.230
FCOS3D-ConvNeXt (Ours)	0.439	0.181	0.217	0.056	0.031	0.374	0.220	0.208	0.455	0.375	0.256

Table 2 further compares these two models with FCOS3D and its ablation experiment across evaluation metrics in

ISSN 2959-6157

the nuScenes dataset.

Table 2 Comparison of Ablation Performance of FCOS3D with FCOS3D-Swin and FCOS3D-ConvNeXt

Backbone	Backbone Size	mAP	NDS
FCOS3D-ResNet101 w/DCN	44.5M [7]	29.8	37.7
above w/ finetune	44.5M [7]	32.1	39.5
above w/ tta	44.5M [7]	33.1	40.3
FCOS3D-Swin (Ours)	28M [8]	23.0	31.9
FCOS3D-ConvNeXt (Ours)	28M [9]	25.6	34.3

Additionally, Figure 2 visualize the inference results of the three models in the same scene.

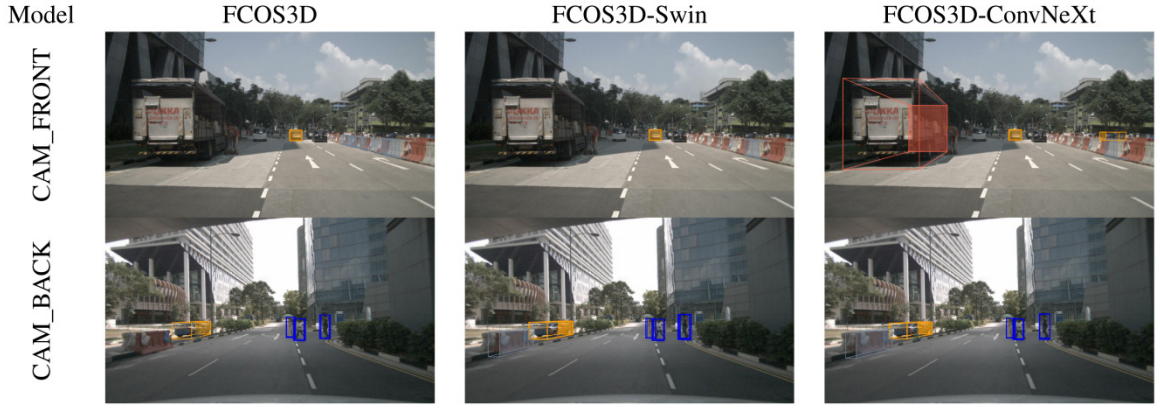

Figure 2 Comparisons of interference results of FCOS3D-Swin and FCOS3D-ConvNeXt from this study and FCOS3D without finetune provided by MMdetection3D officially.

5.2 Discussion of Results

In this study, the performance of the two modified models was compared with the ablation study of FCOS3D [4]. It showed that, although the modified models slightly lagged behind the fine-tuned models in all aspects, they were basically comparable to the baseline of FCOS3D [4], while having smaller parameter sizes. This performance difference can be attributed to the following factors:

· Model parameter size:FCOS3D employs ResNet101 as the backbone, but the Swin Transformer and ConvNeXt used in this study are both tiny architectures with significantly reduced parameter sizes. Despite their impressive performance in 2D detection tasks, the complexity of 3D detection tasks may require larger model perception capacities.

· Data Augmentation Techniques: The study employs the same image augmentation techniques as FCOS3D (scaling and flipping), which might have limited the models' performance compared to techniques originally used in the training of Swin and ConvNeXt, such as random cropping.

· Learning Rate Settings: FCOS3D uses SGD to adjust the learning rate, while Transformer-based models are typically more suitable for AdamW. To maintain consistency, this study did not change this setting, potentially leading to a slower model convergence.

· Training Epochs: The models in this study were only trained for 12 epochs, while FCOS3D [4] underwent both basic training and fine-tuning for a total of 24 epochs. Insufficient training may also have impacted the performance.

5.3 Potential Improvements and Future Work

Based on the existing models, the following improvement directions are proposed:

· Switch to Larger Backbones: Consider using Swin Transformer small or ConvNeXt small, whose parameter sizes are closer to ResNet101, to potentially enhance performance. This act is a trade off between higher computational cost and higher performance of models.

· Enhance Data Augmentation: Introduce the random cropping method used in Swin and ConvNeXt, and explore the Masked Autoencoder technique in ConvNeXt to strengthen the models.

· Adjust Learning Rate Strategies: Switch to the AdamW learning rate adjustment method, which is more suitable

for Transformer architectures.
· Increase Training and Fine-tuning Epochs: By increasing the number of training epochs or conducting fine-tuning, may acquire a better model convergence.
· Integrate Spatio-temporal Information: With the inspiration from subsequent work on FCOS3D, such as PGD [22], performance improvement may achieved by fusing multi-camera information and time-series data.

6. Conclusion

This study evaluates the performance of FCOS-Swin and FCOS-ConvNeXt, modified versions of the FCOS3D model, on the nuScenes dataset. Our findings demonstrate that while the modified models achieve comparable performance to the baseline FCOS3D model, they fall short of the performance achieved by fine-tuned versions of FCOS3D. This study identified several factors that may contribute to this performance gap. Based on these insights, we propose several avenues for future research to further enhance model performance, including:
· Utilizing larger backbone models with stronger feature extraction capabilities.
· Incorporating more effective data augmentation techniques.
· Adopting a more suitable learning rate adjustment method for Transformer-based architectures.
· Increasing the number of training epochs.
· Exploring the integration of temporal and spatial information to improve 3D object detection accuracy.
Overall, this study demonstrates the feasibility of integrating advanced backbone architectures like Swin Transformer and ConvNeXt into the FCOS3D framework for 3D object detection. The findings provide valuable insights into the challenges and opportunities associated with adapting state-of-the-art 2D architectures for 3D object detection tasks.

References

[1] Mao, J., Shi, S., Wang, X., & Li, H. (2023). 3D object detection for autonomous driving: A comprehensive survey. International Journal of Computer Vision, 131(8), 1909-1963.
[2] Singh, A. (2023). Surround-view vision-based 3d detection for autonomous driving: A survey. In 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW).pp. 3235-3244. IEEE.
[3] Zheng, C., Wang, F., Wang, N., Cui, S., & Li, Z. (2024). Towards Flexible 3D Perception: Object-Centric Occupancy Completion Augments 3D Object Detection. https://arxiv.org/abs/2412.05154.
[4] Wang, T., Zhu, X., Pang, J., & Lin, D. (2021). Fcos3d: Fully convolutional one-stage monocular 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 913-922.
[5] Tian, Z., Chu, X., Wang, X., Wei, X., & Shen, C. (2022). Fully convolutional one-stage 3d object detection on lidar range images. Advances in Neural Information Processing Systems, 35, 34899-34911.
[6] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., ... & Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621-11631.
[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770-778.
[8] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012-10022.
[9] Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976-11986.
[10] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580-587.
[11] Sapkota, R., Qureshi, R., Flores-Calero, M., Badgujar, C., Nepal, U., Poulose, A., ... & Karkee, M. (2024). Yolov10 to its genesis: A decadal and comprehensive review of the you only look once series. https://arxiv.org/abs/2406.19407.
[12] Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. https://arxiv.org/abs/2010.11929.
[13] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In European conference on computer vision. pp. 213-229. Cham: Springer International Publishing.
[14] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., ... & Chen, J. (2024). Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16965-16974.
[15] Shi, S., Wang, X., & Li, H. (2019). Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 770-779.
[16] Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp.

ISSN 2959-6157

12697-12705.
[17] Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. Sensors, 18(10), 3337.
[18] Chen, Y., Liu, S., Shen, X., & Jia, J. (2020). Dsgn: Deep stereo geometry network for 3d object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12536-12545.
[19] Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., ... & Dai, J. (2022). Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In European conference on computer vision. pp. 1-18. Cham: Springer Nature Switzerland.
[20] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF international conference on computer vision. pp. 6569-6578.
[21] Liu, Z., Wu, Z., & Tóth, R. (2020). Smoke: Single-stage monocular 3d object detection via keypoint estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 996-997.
[22] Wang, T., Xinge, Z. H. U., Pang, J., & Lin, D. (2022). Probabilistic and geometric depth: Detecting objects in perspective. In Conference on Robot Learning. pp. 1475-1485. PMLR.
[23] Li, P., Chen, X., & Shen, S. (2019). Stereo r-cnn based 3d object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7644-7652.
[24] Philion, J., & Fidler, S. (2020). Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 194-210. Springer International Publishing.
[25] Huang, J., Huang, G., Zhu, Z., Ye, Y., & Du, D. (2021). Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. https://arxiv.org/abs/2112.11790.
[26] Wang, Y., Guizilini, V. C., Zhang, T., Wang, Y., Zhao, H., & Solomon, J. (2022). Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In Conference on Robot Learning. pp. 180-191. PMLR.
[27] Vora, S., Lang, A. H., Helou, B., & Beijbom, O. (2020). Pointpainting: Sequential fusion for 3d object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4604-4612.
[28] Huang, T., Liu, Z., Chen, X., & Bai, X. (2020). Epnet: Enhancing point features with image semantics for 3d object detection. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. pp. 35-52. Springer International Publishing.
[29] Pang, S., Morris, D., & Radha, H. (2020). CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10386-10393. IEEE.