Proceedings of the 5th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/134/2025.22252

Comparison of Different Object Detection and Classification Models

Qian Shan[1*], Qun Yang[2], Jinjiang Ding[3], Yanbin Hou[4], Kaiming Gu[5], Zitong Sun[6], Jiahui Zhou[7]

[1]School of Letter and Science, University of California, Davis, USA
[2]Yangzhou High School, Yangzhou, China
[3]North London Collegiate School Singapore, Singapore
[4]Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK
[5]International Institute of Technology, Xi'an University of Technology, Xi'an, China
[6]Department of Statistical Science, University College London, London, UK
[7]Computer Science, Shandong University, TsingTao, China
**Corresponding Author. Email: Karl20011109@gmail.com*

Abstract: Object detection has become an important task in the field of computer vision, where the goal is to recognize and classify objects in images or videos. This paper presents a comparative analysis of different object detection models, focusing on convolutional neural networks (CNN) and transformer-based architectures. CNN-based models (e.g., the YOLO family) have made significant progress in real-time object detection by efficiently extracting local features via convolutional operations. In contrast, transformer-based models, such as the Visual Transformer (ViT), use self-attention mechanisms to capture global dependencies, improving performance on large-scale datasets. This research explores the evolution of these models and examines their foundations, strengths, and weaknesses. Through experimental evaluations, we show that CNNs continue to dominate when data and computational power are limited, while Transformers exhibit superior scalability and accuracy in complex environments. Our results highlight the complementary nature of these approaches and emphasize the need for hybrid models to achieve optimal performance in different object detection tasks.

Keywords: Transformer, Convolutional Neural Network, Object Detection, Image Classification.

1. Introduction

Object detection and image classification are two of the most significant tasks in the computer vision field. Object detection mostly focuses on identifying the location and boundaries of different objects according to images and videos, at the same time classifying them into different categories. While the image classification typically pertains to single-object images. It mostly focuses on assigning a label to an entire image and classifying them into typical categories. Because of the different appearances of various objects and factors such as lighting and occlusion occurs while imaging, object detection, and image classification have always been the most challenging part of the field of computer vision.

DOI: 10.54254/2755-2721/134/2025.22252

Over recent years, the Convolutional Neural Network (CNN) methods have become widely used in the field, methods of CNN are generally divided into two categories: two-stage algorithms and one-stage algorithms [1]. Two-stage algorithms first generate a rectangular region, which is called a region proposal (RP), serving as a bounding box for the object of interest, followed by classification using a CNN. On the other hand, one-stage algorithms bypass the region proposal step, directly extracting features from the network to predict both the classification and location of objects.
With the continuous development in the field of computer vision, algorithms for object detection and image classification are constantly updating and iterating. For instance, the RCNN series, a prominent example of two-stage algorithms, has seen continuous development. Although some of its disadvantages, such as the time required for feature extraction, have been solved roughly, its real-time performance remains poor, preventing it from reaching the goal of real-time object detection. Therefore, the paper focuses on YOLOv5 and v8 from the YOLO series. These models eliminate the need for the two-stage process found in RCNN and instead return object locations and classifications directly at the output layer.

While the CNN-based algorithms have long held a dominant position in object detection and image classification, the introduction of Transformer models marked a significant shift in this field. Ever since their introduction, a variety of Transformer models have been applied to object detection and image classification, offering advantages such as scalability to larger datasets and the ability to capture global features through self-attention mechanisms. In contrast, CNNs excel at extracting local features through convolution operations, making them computationally efficient and more suitable for smaller datasets and resource-constrained environments.

This paper provides an overview of the CNN and vision transformer models we use for object detection and image classification, compares their performance across datasets of varying sizes, and draws corresponding conclusions.

2. Literature Review

CNN is a widely used model for image and video processing tasks. It was first proposed by Yann et al. in 1989 for handwritten digit recognition, and the first successful application of this model to practical problems led to the birth of LeNet-5. As the basic CNN model, LeNet-5 consists of two convolution layers, two pooling layers, and three fully connected layers [2]. In 2012, Alex Krizhevsky et al. proposed a new model (AlexNet) based on CNN that significantly improves performance by combining ReLU activation functions, massive enhancement [3], and Dropout. Over the next decade, many other models were proposed. For example, in 2014, the Visual Geometry group at the University of Oxford proposed VGGNet, which has deeper convolutional networks, using smaller (say 3*3) convolutional cores in each layer to capture more detail [4]. Also, in 2014, Google introduced GoogLeNet [5], which incorporated an Inception module for parallel processing through multi-scale convolution (i.e., convolution cores of different sizes), which improved feature extraction rates and reduced the number of parameters. In addition, the model adopts a global average pooling layer to reduce the dependence on the fully connected layer and reduce the overall complexity.

In 2015, He Keming et al. proposed ResNet. It uses the residual module to introduce skip connections to solve the problem of gradient disappearance and network degradation caused by depth increase [6]. The addition of residual connections makes it possible to train deeper and more complex neural networks successfully. Joseph Redmon et al. proposed You Only Look Once (YOLO) in 2016. It is a real-time object detection algorithm that uses a single neural network to predict bounding boxes and class probabilities. Compared with the traditional two-stage detection algorithm, it is faster and more efficient. After continuous optimization, it has become one of the most effective algorithms widely used in object detection [7-10]. DenseNet and MobileNet were both proposed in 2017. DenseNet uses dense connections to connect the output of each layer with the output of all previous

DOI: 10.54254/2755-2721/134/2025.22252

layers, improving feature transmission and reuse, reducing the number of parameters, and improving performance [11]. MobileNet uses depth-separable convolution to reduce computational complexity and number of parameters [12].

Vaswani et al. proposed the transformer model in 2017 [13]. The main innovation is the self-attention mechanism, which allows the model to measure the importance of different words in a sentence without focusing on the relationships in the sequence.

In 2018, Bidirectional Encoder Representation (BERT) was proposed by Google, and in the same year, OpenAI published GPT [14]. From 2018-2020, OpenAI kept iterating the model and published the GPT-1, GPT-2, GPT-3 and GPT-4 models [15-17]. In 2020, Google proposed the Text-to-text Transfer Transformer (T5) model, which treats all NLP tasks as a text-to-text problem and has achieved excellent results and demonstrated excellent flexibility and powerful performance [18]. After this, the researchers began applying Transformers to computer vision tasks and co It excelled in a variety of visual tasks. Some researchers have begun to explore the integration of CNN and the Transformer model, hoping to combine the advantages of the two methods and propose a model that can solve more complex visual tasks [19, 20].

3. Different Object Detection Models

This paper will delve into the realm of object detection, focusing on two primary model architectures: CNN-based and transformer-based. We will explore several representative models from each category, elucidating their underlying principles, strengths, and weaknesses. Comparative analyses will be conducted to highlight key differences and similarities.

3.1. CNN-Based Models

Convolutional Neural Networks (CNNs), a type of feedforward neural network inspired by the biological processes of natural vision, have become a focal point of research across many scientific disciplines. Their ability to directly input raw images, bypassing the need for complex pre-processing, has made CNNs especially popular in object detection. A simple CNN model consists of multiple convolutional layers, ReLU layers, and pooling layers. Images pass through these layers, and their features are extracted. Finally, these features are fed into a fully connected layer. Each feature map can be viewed as a matrix of neurons, like the neurons in a backpropagation neural network. Figure 1 shows the structure of a simple CNN model.

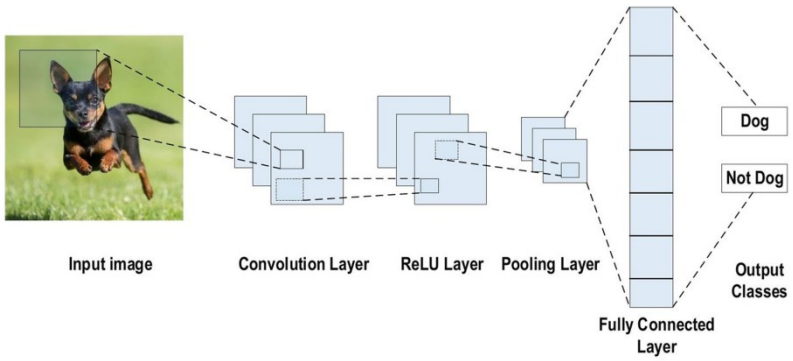

Figure 1: The structure of a simple CNN model [21].

When an image is input, the convolutional kernel slides over the image in the convolutional layer, performing a weighted sum of local regions to extract image features such as edges and textures. The ReLU layer performs a non-linear transformation on the output of the convolutional layer, enhancing the network's expressive power. The pooling layer downsamples the feature map, reducing the

DOI: 10.54254/2755-2721/134/2025.22252

number of parameters and preventing overfitting while preserving the main features. Common pooling methods include max pooling and average pooling. The output of the pooling layer is flattened and connected to a fully connected layer. This fully connected layer acts similarly to a traditional neural network, mapping the learned features to different classes. Through these processes, CNNs can learn the hierarchical features of an image, from low-level edges and textures to high-level semantic information.

3.1.1. YOLO Series

The You Only Look Once (YOLO) algorithm is a representative one-stage algorithm. YOLO's structure is similar to the classification CNN models, which means it is not very complicated. The biggest difference between YOLO and other classification CNN models is that YOLO uses a linear function at the output layer, so YOLO can not only classify objects but also show their location. After long-term development, the YOLO algorithm has been very mature. Figure 2 shows the evolution of the YOLO algorithm from 2015 to 2023.

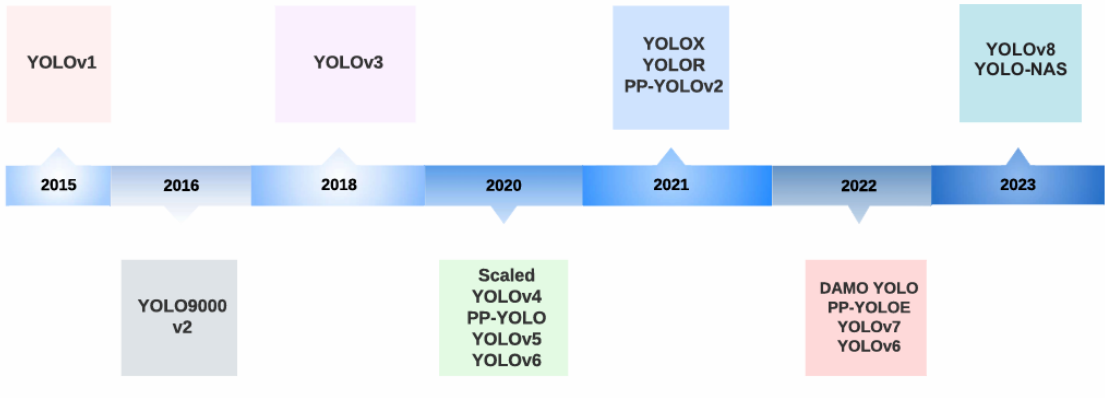

Figure 2: The timeline of YOLO versions [22].

3.1.2. The Structure of Different YOLO algorithm

This section will delve into two prominent models within the YOLO family: YOLOv5 and YOLOv8. YOLOv5 rapidly established itself as a benchmark in the field of object detection after being published because of its exceptional performance, user-friendly interface, and thriving community. These attributes make it a fitting representative of the overall capabilities of the YOLO series. YOLOv8, a more contemporary iteration of YOLO, introduces enhancements in terms of performance and functionalities. Nevertheless, for applications demanding high real-time performance and limited datasets, YOLOv5 might behave better.

YOLOv5 is a state-of-the-art object detection model introduced in 2020 by Glenn Jocher, just a few months after the release of YOLOv4. It has gained significant recognition for its exceptional speed and accuracy. While largely inheriting many improvements from YOLOv4, YOLOv5 is developed using PyTorch instead of Darknet. It excels at rapidly identifying and localizing multiple objects within an image. The combination of its efficient architecture and advanced techniques enables real-time performance while maintaining impressive detection capabilities. Figure 3 shows the structure of YOLOv5.

For modern image detection models, we usually divide them into Backbone, Neck and Head. The architecture employs a modified CSPDarknet53 as the backbone network, which includes a Stem layer followed by convolutional layers to extract image features. Spatial Pyramid Pooling Fast (SPPF)

DOI: 10.54254/2755-2721/134/2025.22252

accelerates computation by pooling features into fixed-size maps. Each convolutional layer incorporates batch normalization and SiLU activation functions. The network's neck utilizes SPPF and a modified CSP-PAN structure, while the head is similar to YOLOv3.YOLOv5 uses multiple data augmentation methods during training, such as Mosaic augmentation, random affine transforms, MixUp, HSV color space augmentation, and random horizontal flips. To cater to different applications and hardware constraints, YOLOv5 provides five models of varying sizes: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x.

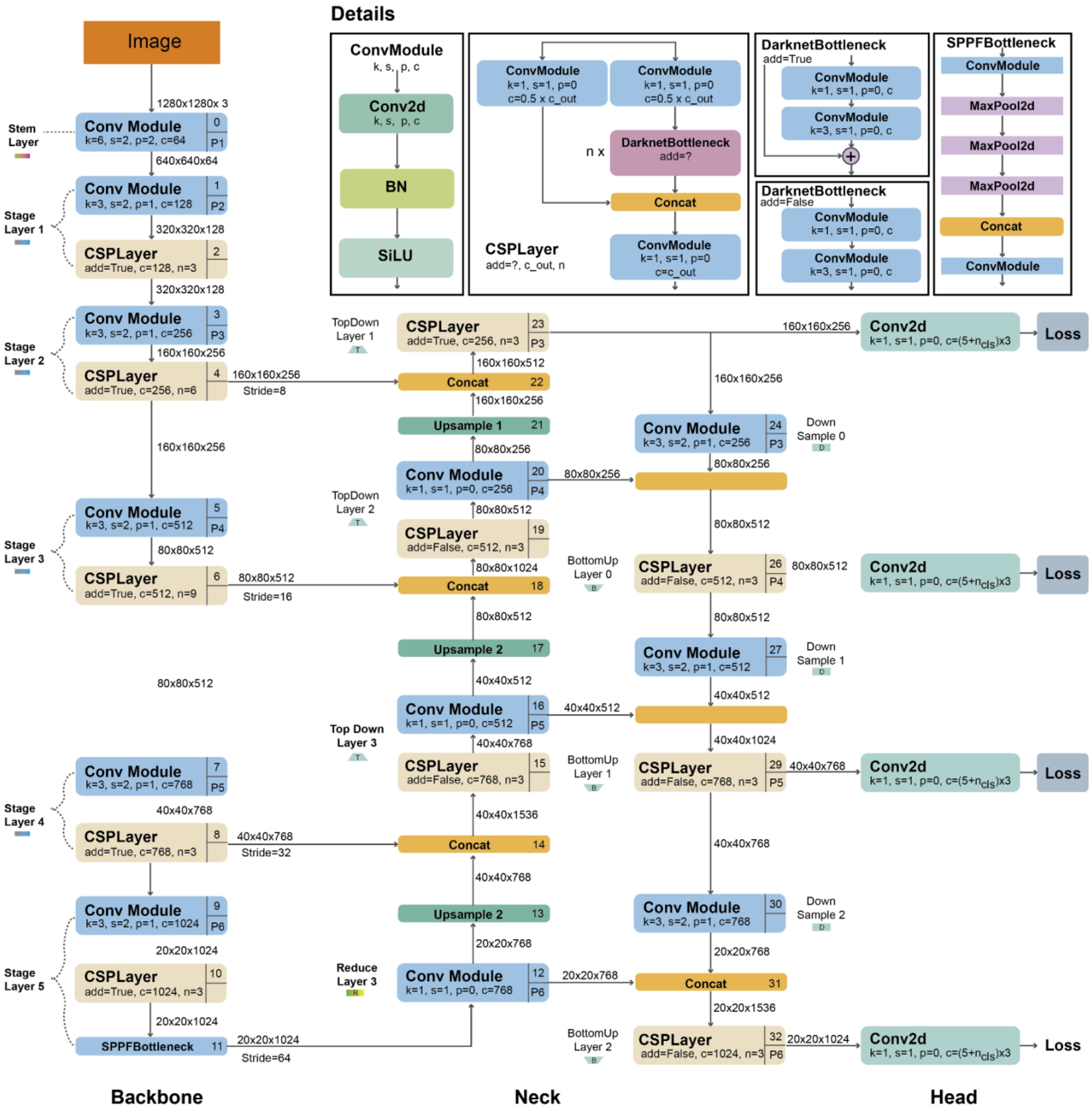

Figure 3: The Structure of YOLOv5 [23].

YOLOv8 (You Only Look Once version 8), the loss function of the YOLOv8 framework consists of three main major terms: Classification Loss term, Confidence Loss term, and Bounding Box Regression Loss term. Each of these components works toward achieving the same goal: minimizing

DOI: 10.54254/2755-2721/134/2025.22252

model parameters and enhancing the overall model performance [24]. Below is a detailed breakdown of each piece of the puzzle: Classification Loss is the evaluation method that deals with the difference between the predicted and the actual categories. In the process of target detection, the classification loss must identify the object as the target and furnish precise classification categories of the acquired targets. The classification loss comprises the Cross-Entropy Loss function, primarily used to measure the distance between the predicted probability distribution and the truth distribution to classify the objects correctly. Confidence Loss Plays the confidence of the model that detected the object by the bounding box. Indicates whether the goal exists within the predicted bounding box, which is related to the goal score. In object detection, the Confidence Loss is usually taken as Binary CrossEntropy for the training data which is an important loss function for the model to help it predict whether a target is likely to be present in a certain specific grid cell. The Bounding Box Regression Loss score calculates the differences between the boundary box prediction and the actual positioning and validates that the positioning, size, and shape are all precisely positioned. Bounding Box Regression Loss is the way of locating with the highest precision of objects by reducing the proportion between the predicted boundary box and the actual object.SIoU (Skew Intersection over Union) Loss Function is a method of Loss Function design of point target detection in the bounding box regression. They are to optimize the prediction box so that its position is correct, the scale is appropriate, and their shapes are similar to increase the detecting rate. IoU (Intersection over Union) measures the ratio of how similar the demon box is to the real box [25].

Improved Convergence: Superior Convergence In the case of object detection, the IoU factor is numerically concerned with the overlap of two boxes, whereas SIoU is annetable to more sufficiently take into account the direction and contour shape of the object, making a better performance of the system when researcher encounters a case that researcher need a tighter box for instance. IoU only measures the overlap between the prediction and truth boxes. It provides no useful information for non-overlapping cases. SIoU comprehensively considers the spatial relationship and scale of bounding boxes to provide richer information and accurate target positioning so as to predict bounding boxes more accurately.SIoU is suitable for a wide range of target detection tasks. SIoU is more robust to changes in the target size and aspect ratio, and it can efficiently handle different object shapes and sizes to improve the accuracy of bounding box prediction [26]. The dataset is selected from the basic vehicle dataset, a total of 250 pictures, before and after the comparison, the former is for the improvement of the basic YOLOv8 model, the latter is to improve the SIou function after the YOLOv8 model. The accuracy value is improved from 0.494 to 0.540, and the curve is smoother for the result. Figure 4 shows the specific data curves for model YOLOv8 after 100 epochs of training. Moreover, Figure 5 shows the specific data curves after 100 epochs of training for model YOLOv8 with improved SIou loss function.

DOI: 10.54254/2755-2721/134/2025.22252

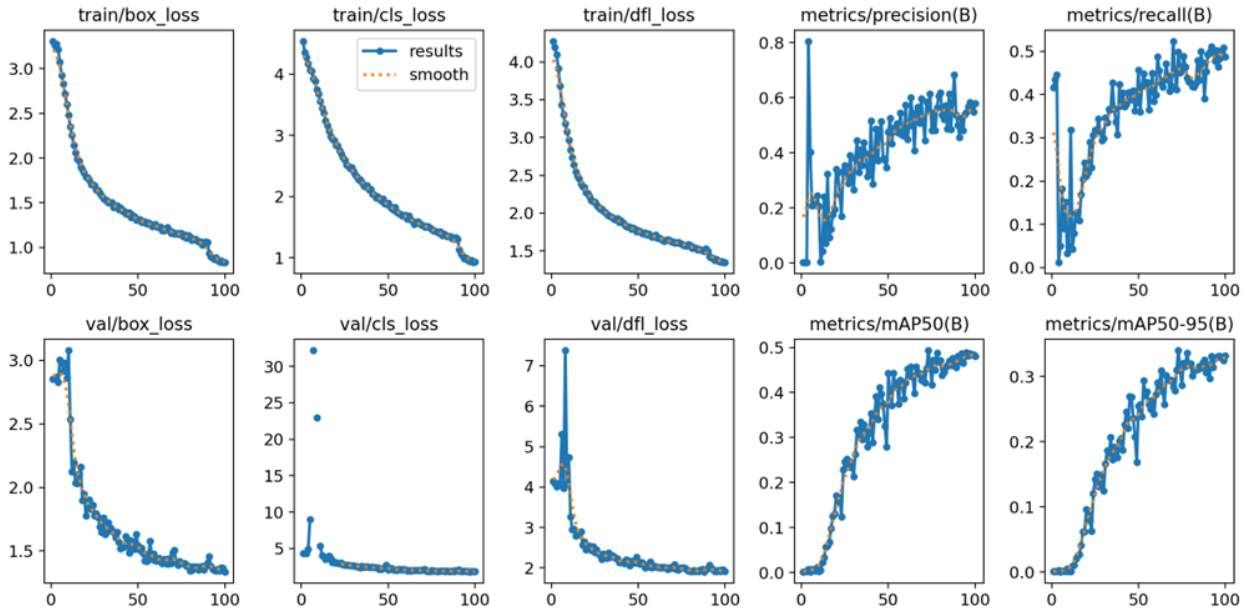

Figure 4: Specific data curves for model YOLOv8 after 100 epochs of training.

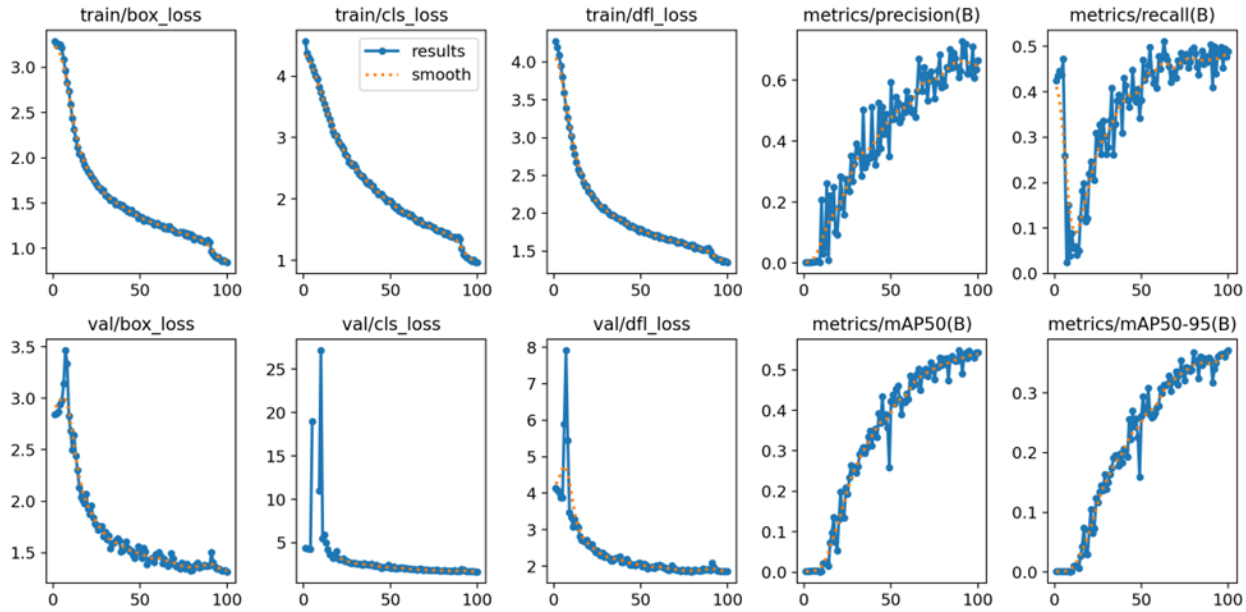

Figure 5: Specific data curves after 100 epochs of training for model YOLOv8 with improved SIou loss function.

3.2. Transformer-Based Models

Vision Transformer (ViT) is an innovative model introduced in the field of computer vision in recent years. Unlike traditional convolutional neural networks (CNNs), ViT is based on the Transformer architecture and was originally designed for natural language processing tasks. The core idea of ViT is to divide the input image into fixed-size blocks, flatten them into a vector, and input them into the Transformer network in a sequence.

The Transformer encoder consists of a multi-layer self-attention mechanism and a feedforward neural network. The self-attention mechanism can effectively capture the global dependencies

DOI: 10.54254/2755-2721/134/2025.22252

between image blocks, thereby obtaining more comprehensive image information. After being processed by the multi-layer Transformer encoder, the resulting feature vector can be used for a variety of computer vision tasks, such as image classification and object detection. Figure 6 shows the overall structure of the Vit model. The advantage of ViT lies in its powerful global modeling ability, which enables it to learn complex features of images without relying on convolution operations. Research has shown that ViT performs well on large-scale datasets, especially with sufficient data, and can outperform traditional CNN models [27]. However, it has high data and computing resource requirements, so practical applications usually require large-scale training data and computing power.

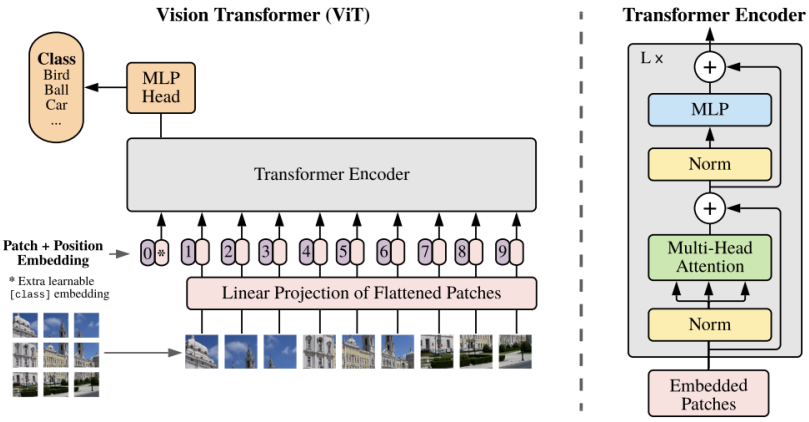

Figure 6: Model overview.

We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable “classification token” to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. [27].

The Transformer encoder plays a crucial role in natural language processing tasks. It is composed of multiple layers, each containing four key components:

1. The input embedding layer converts the token into a dense vector and incorporates positional encoding to preserve the token's order.

2. The multi-head self-attention mechanism calculates the attention score for each pair of tokens in the sequence, allowing the model to focus on different parts of the sequence. Multiple attention heads run in parallel to learn different aspects.

3. The feedforward neural network (FFN) applies two linear transformations with ReLU activation to independently process the representation of each token.

4. Residual connections and layer normalization are connected to each sub-layer (self-attention and feedforward network), which helps stabilize training and improve convergence speed.

These components are stacked in multiple layers, with the output of one layer serving as the input for the next layer. The final output of the encoder consists of a series of vectors that capture contextual information for each token, making it suitable for various NLP tasks such as translation, text summarization, and classification.

DOI: 10.54254/2755-2721/134/2025.22252

4. The Comparison of Different Models

4.1. Comparison between YOLOv5 and YOLOv8 models

In the same dataset, YOLOv5 and YOLOv8 have different test results, for the selected dataset, the computer uses the YOLOv5n and YOLOv8n models to compare with each other. For the recognition effect for selected card data, YOLOv5n's accuracy is better than YOLOv8n's accuracy, computer ran 8 epochs of training on YOLOv5n and YOLOv8n each using a dataset containing about 4000 images of different kind of cards. Figure 7 shows the first image shows the test results of the YOLOv5n. Moreover, Figure 8 shows the second image shows the test results 0f the YOLOv8n. Finally, the accuracy of YOLOv5n is 0.982, and the accuracy of YOLOv8n is 0.978.

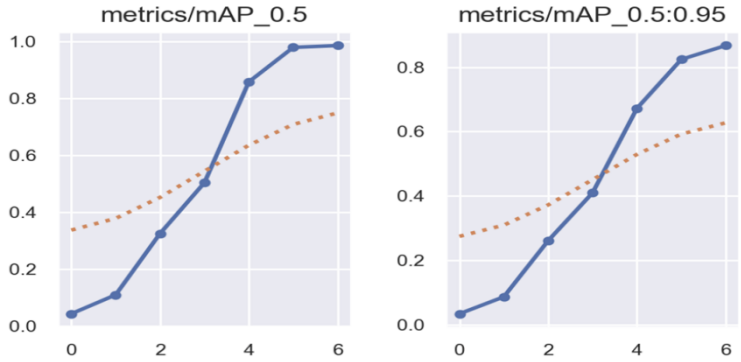

Figure 7: The first image shows the test results of the YOLOv5n.

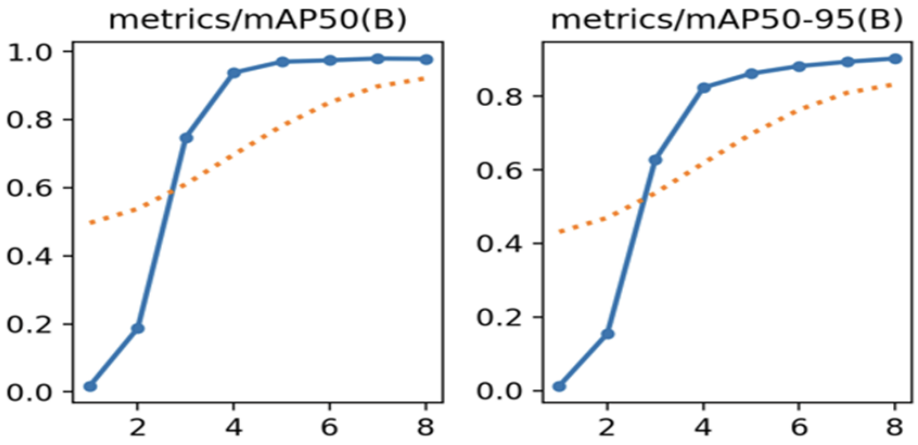

Figure 8: The second image shows the test results of the YOLOv8n.

4.2. Comparison between CNN and Transformer models

Using the same dataset for recognizing handwritten digits 0-9, the CNN image classification model and the vision transformer image classification model were trained separately, and the resulting data is in Figure 9 and Figure 10. In Figure 9, the training loss of Vision Transformer (ViT) in the first few epochs dropped rapidly from 2.00 to about 0.18, showing that the model quickly adapted to the data features. The training loss and validation loss stabilized from the third epoch and were below 0.1. This shows that the model has stabilized, and further training will mainly be for optimization and fine-tuning. The validation loss is not much different from the training loss during the entire training process and is even lower in some epochs, showing that ViT is very stable.

DOI: 10.54254/2755-2721/134/2025.22252

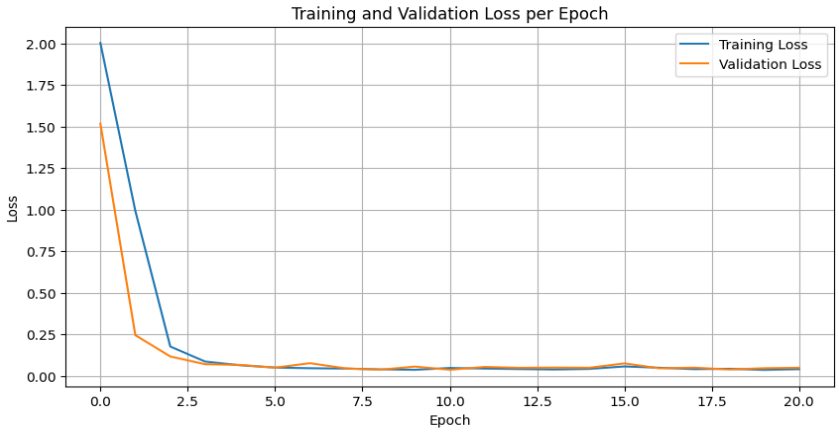

Figure 9: The graph of training and validation loss per epoch of ViT.

Figure 10 shows the training data of the CNN model image classification. Unlike ViT, the training loss of CNN slowly and continuously decreases from 0.1659 to 0.0075 in the 10th epoch. This progressive mode has certain advantages in extracting local features of images. The training ended early at epoch 12, probably because the model had begun to perform poorly on the validation set, reflecting that the learning process might need more tuning and optimization. In terms of validation loss fluctuation, compared with ViT, the validation loss of the CNN model shows more fluctuations in the reduction process. Especially in the later stage, the validation loss even has an upward trend, which may be a sign that the model is overfitting on specific data.

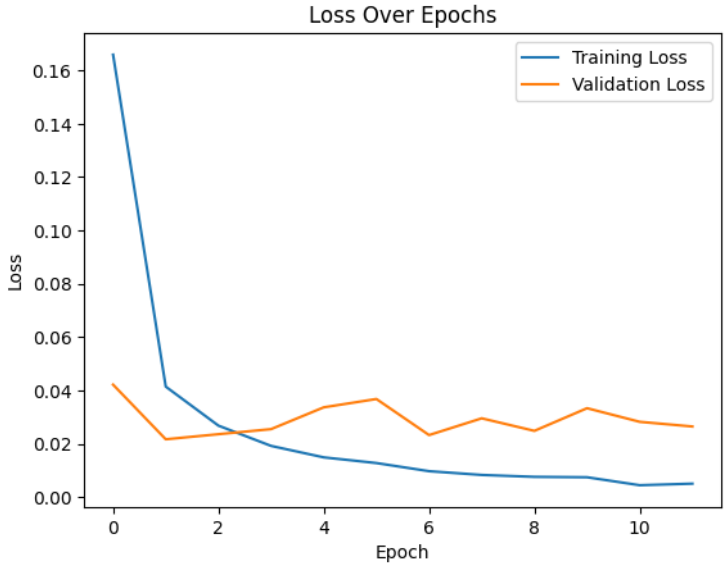

Figure 10: The graph of training and validation loss per epoch of CNN.

According to the comparison of the two figures, ViT's fast learning and excellent generalization abilities are more prominent than CNN's. However, ViT requires more computing resources during training due to its higher complexity. Loss reduction may also be unstable, and we can see slight fluctuations in Figure 9. Compared with ViT, CNN shows its shortcomings of slow convergence and requiring more cycles to achieve optimal performance. Secondly, as shown in Figure 10, CNN has the risk of overfitting. However, because CNN has mature technology and framework, it is usually easier to optimize and is widely tested in image tasks. By comparing the data, we can clearly feel the

DOI: 10.54254/2755-2721/134/2025.22252

advantages and disadvantages of ViT and CNN. In general, each model has its unique advantages and limitations. Understanding these can help better use these models to solve practical problems.

Table 1: Model performance evaluation comparison.

Model	Accuracy	Precision	Recall	F1 Score
ViT	0.9911	0.9912	0.9910	0.9911
CNN	0.9913	0.9912	0.9912	0.9912

When dealing with handwritten digit recognition tasks, particularly on simple datasets like the classic MNIST dataset or similar ones, models tend to perform exceptionally well, often nearing perfect accuracy. The task's relative simplicity allows even basic models to achieve high-performance metrics. Table 1 shows how the CNN model and the Vit model perform differently under the same task. In comparing the performance of ViT and CNN on such datasets, both models exhibit very close results. While CNN slightly outperforms ViT in terms of precision, recall, and F1 score, the differences are minimal. This slight edge suggests that CNN might be marginally better suited for this specific dataset. However, these differences might hold little practical significance given the negligible gap. Therefore, the choice between ViT and CNN should also consider other factors, such as model complexity, computational resource requirements, and inference speed, in addition to their performance metrics. It is worth noting that ViT, a slightly complex architecture, can still be comparable to classic models designed specifically for such tasks when processing relatively simple datasets, which is undoubtedly a reflection of its strong adaptability in the field of image recognition.

5. Conclusion

This paper provided a comprehensive overview and comparison of different object detection models, specifically focusing on CNN-based and transformer-based approaches. Through analyzing prominent implementations within each category, as well as experimental evaluations on various datasets, we gained valuable insights into the strengths and weaknesses of these model architectures. CNNs, represented by the YOLO family of models, have achieved tremendous success in computer vision tasks through their ability to efficiently extract hierarchical local features via convolutional operations. YOLOv5 and YOLOv8 demonstrated state-of-the-art one-stage object detection performance through advancements like efficient backbones, multi-scale feature fusion techniques, and novel loss functions. However, CNNs can struggle with larger, more complex datasets due to their reliance on local connectivity patterns.

In contrast, transformer models like Visual Transformers leverage self-attention to capture long-range dependencies, empowering them to learn comprehensive representations of images through global feature interactions. ViT exhibited superior scalability and classification accuracy compared to CNNs on large-scale datasets, demonstrating transformers' potential. Nevertheless, transformer-based vision models demand significantly more computational resources and data to train compared to CNNs.

Our experimental results show that while CNNs may achieve competitive results with limited data and hardware, transformers scale favorably to tackle larger, more challenging vision tasks. However, CNNs remain preferable for applications with constrained resources like mobile and embedded systems. Moving forward, the complementary strengths of CNNs and transformers indicate that hybrid convolutional transformer architectures may achieve optimal performance across different scenarios.

DOI: 10.54254/2755-2721/134/2025.22252

To sum up, this research provides valuable insight into the evolution of modern object detection models. Both CNNs and transformers play important roles and will likely continue co-developing computer vision.CNN technology will further optimize efficiency, while transformers aim for richer global representations. Future work should explore techniques like multimodal fusion to combine local and holistic pattern learning. Underlying these advancements is deep learning's ability to continually push the boundaries of vision understanding through ever more powerful yet efficient model designs.

Acknowledgement

Qian Shan, Qun Yang, Jinjiang Ding, Yanbin Hou, Kaiming Gu, Zitong Sun, and Jiahui Zhou contributed equally to this work and should be considered co-first authors.

References

[1] *Hou-I Liu, Yu-Wen Tseng, and Kai-Cheng Chang, et al. (2024), A DeNoising FPN With Transformer R-CNN for Tiny Object Detection. DOI: https://doi.org/10.48550/arXiv.2406.05755*
[2] *Lecun, Y. et al. (1998) 'Gradient-based learning applied to document recognition', Proceedings of the IEEE, 86(11), pp. 2278–2324. doi: 10.1109/5.726791.*
[3] *Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017) ImageNet classification with deep convolutional neural networks, Communications of the ACM. New York: Association for Computing Machinery, pp. 84–90. doi: 10.1145/3065386.*
[4] *Simonyan, K. and Zisserman, A. (2015) 'Very Deep Convolutional Networks for Large-Scale Image Recognition', arXiv.org. doi: 10.48550/arxiv.1409.1556.*
[5] *Szegedy, C. et al. (2014) 'Going Deeper with Convolutions', arXiv.org. doi: 10.48550/arxiv.1409.4842.*
[6] *Kaiming He et al. (2016) 'Deep Residual Learning for Image Recognition', in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 770–778. doi: 10.1109/CVPR.2016.90.*
[7] *Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).*
[8] *Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).*
[9] *Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.*
[10] *BochNovsNiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, pp.5-10.*
[11] *Huang, G. et al. (2018) 'Densely Connected Convolutional Networks', arXiv.org. doi: 10.48550/arxiv.1608.06993.*
[12] *Howard, A. G. et al. (2017) 'MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications', arXiv.org. doi: 10.48550/arxiv.1704.04861.*
[13] *Vaswani, A. et al. (2023) 'Attention Is All You Need', arXiv.org. doi: 10.48550/arxiv.1706.03762.*
[14] *Devlin, J. et al. (2019) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', arXiv.org. doi: 10.48550/arxiv.1810.04805.*
[15] *Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I., 2018. Improving language understanding by generative pre-training.*
[16] *Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8), p.9.*
[17] *Brown, T.B., 2020. Language models are few-shot learners. arXiv preprint ArXiv:2005.14165.*
[18] *Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140), pp.1-67.*
[19] *X. Chu et al., "Conditional positional encodings for vision trans[1]formers," 2021, arXiv:2102.10882.*
[20] *H. Wu et al., "CVT: Introducing convolutions to vision trans[1]formers," 2021, arXiv:2103.15808*
[21] *Dosovitskiy, A. et al. (2020) An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. DOI: https://doi.org/10.48550/arXiv.2010.11929.*
[22] *Alzubaidi L, Zhang J, Humaidi A J, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions[J]. Journal of Big Data, 2021, 8: 1-74. DOI: https://doi.org/10.1186/s40537-021-00444-8*

DOI: 10.54254/2755-2721/134/2025.22252

[23] Terven J, Córdova-Esparza D M, Romero-González J A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas[J]. Machine Learning and Knowledge Extraction, 2023, 5(4): 1680-1716.DOI: https://doi.org/10.3390/make5040083

[24] Zhang, Lin, et al. "Research on improved YOLOv8 algorithm for insulator defect detection." Journal of Real-Time Image Processing 21.1 (2024): 22.

[25] Li, Xin, et al. "Coal mine belt conveyor foreign object detection based on improved yolov8." 2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC). Vol. 11. IEEE, 2023.

[26] Wang, Yan, et al. "An improved YOLOv8 algorithm for rail surface defect detection." IEEE Access (2024).

[27] Dosovitskiy, A. et al. (2021) 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale', arXiv.org. doi: 10.48550/arxiv.2010.11929.