## 1 Introduction

Object detection, a fundamental task in computer vision, plays a vital role in various applications such as autonomous driving, surveillance, and image understanding. With the rapid development of deep learning techniques, convolutional neural networks (CNNs) such as Residual Network [1], have demonstrated remarkable performance in object detection tasks. Among them, YOLO (You Only Look Once) series, characterized by its real-time processing capability and high accuracy, has gained widespread attention in recent years.

The PASCAL VOC (Visual Object Classes) datasets, encompassing VOC2012 and VOC2007, represent pivotal resources in the realm of computer vision, serving as benchmarks for object detection tasks [2]. These datasets comprise a diverse array of images spanning numerous object categories, providing researchers with data for model training and evaluation.

The PASCAL VOC datasets present several challenges for object detection models. These include three main challenges: class imbalance, varied object scales, and uneven distribution of high-level semantic features. Addressing these challenges is crucial for improving the performance of object detection models like YOLOv8 on these datasets.

Our workloads and contributions can be summarized as follows:

1. Comprehensive analysis is conducted on the VOC2012 and VOC2007 datasets, identifying key challenges for object detection tasks.
2. Entropy-based Sampling is implemented to tackle the scarcity of samples within certain categories.
3. YOLOv8-NSK model is proposed based on attention mechanisms. Firstly, modifications are made to the C2f component by integrating the Selective Kernel Networks (SKNets) [3] to enhance the model's recognition performance of muti-scale objects. Secondly, adjustments are applied to the neck component through the incorporation of the Normalization-based attention module (NAM) [4] to improve the model's ability to identifying high-level semantic features and small objects, while suppressing noise features.
4. Extensive experiments, including comparative and ablation studies, are conducted to validate the effectiveness of the proposed methods. In the comparative experiments, different attention mechanisms are compared when added to the neck component of YOLOv8, selecting the one that yields the highest accuracy improvement. Subsequently, the impact of adding the same attention mechanism to different positions within the backbone, C2f modules and neck components is assessed. Furthermore, the enhanced YOLOv8 model is compared with other baseline models to illustrate the superiority of YOLOv8NSK. In the ablation experiments, the performances of SKNets and NAM are evaluated to assess the contribution of each attention module. With regard of the baseline model, the mAP of YOLOv8-NSK achieved improvement of 1.2%.

## 2 Literature review/related works

The fundamental tasks in computer vision can be classified as following: classification, localization, object detection, instance segmentation, as shown in Figure 1. All the tasks take an input (or a frame in video) and return outputs of different kinds with respect to different task objectives. The task of "finding what objects are where" consists of two subtasks: classification and localization, focusing on learning which class the object belongs to in the classes of interest and how to localize it with a bounding box, i.e. a rectangle restricting the outline of the object, respectively.

Object detection has long been the fundamental task in the domain of computer vision. It dated way back to 1990s when deep learning was yet to be recognized generally by the field of research. Some traditional object detection methods such as ViolaJones object detection method [5], Histogram of gradient [6], Deformable parts model [7] were developed around 2000s. However, traditional methods for object detection face several limitations, in this situation deep learning-based approaches which can effectively address the challenges posed by real-world images were introduced after the birth of AlexNet[8] . 2014 marked a turning point in object detection with the groundbreaking work of Girshick et al [9] whose approach of R-CNN leads to an exceptional pace of development in the years that followed.

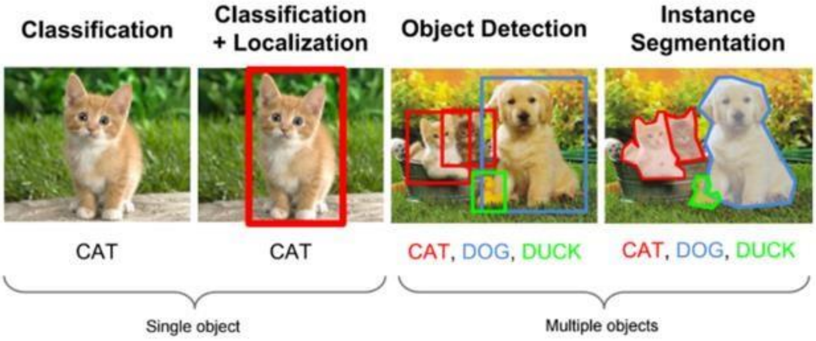


**Figure 1.** Taxonomy of computer vision tasks

Two primary categories of detectors have emerged in the deep learning field, one of which is two-staged detectors, including RCNN [9],SPPNet [10],Fast RCNN [11], which first generate region proposals and then classify each proposal, another is one-stage detectors represented by YOLO series, SSD [12] and RetinaNet [13], which directly predict object bounding boxes and class labels from an image in a single step. The choice of one-stage or two-stage detector depends on the specific application. One-stage detectors are preferred when speed is more important than accuracy, while two-stage detectors are preferred when accuracy is more important than speed.

The shift towards Transformer-based models in object detection was initiated by the introduction of DEtection TRansformer (DETR) [14]. DETR simplified the object detection process by utilizing a Transformer encoder-decoder network [15] to resolve issues in NLP task, eliminating the need for traditional region proposal and post-processing methods like non-maximum suppression (NMS). Modifications such as Deformable DETR [16], Swin Transformer [17], DINO [18] and RealTime Detection Transformer (RT-DETR) [19] have been developed to address the limitations of the original DETR while maintaining or improving computational efficiency and detection accuracy.

At the same time, several methods can be used to optimize the performance of the baseline model, like spatial pyramid pooling, feature fusion and attention mechanism. SPPNet [10] enables CNNs to process images of arbitrary size and ratio by dividing input feature maps into equally sized sub-regions and performing pooling operations within each, enhancing both performance and efficiency in testing. T.-Y Lin et al proposed Feature Pyramid Network (FPN) [20], a top-down architecture with lateral connections to fuse low-res and semantically strong features with high-res and semantically weak features together. Attention mechanisms can be inserted into baseline module, allowing neural networks to selectively focus on certain parts of input data, enhancing model performance by assigning varying degrees of importance to different elements.

## 3 Data analysis

### 3.1 PASCAL VOC (Visual Object Classes) datasets

Image annotation of PASCAL VOC2007 and VOC2012 dataset include the following attributes for every object in the target set of 20 object classes:

**class**: one of: aeroplane, bird, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, tv/monitor.

**bounding box**: an axis-aligned bounding box surrounding the extent of the object visible in the image.

**Table 1.** Statistics of the VOC2007 and VOC2012 trainval and test datasets

| | Trainval | | Test | |
|---|---|---|---|---|
| | Images | Objects | Images | Objects |
| VOC 2007 | 5,011 | 15,662 | 4,952 | 14,976 |
| VOC 2012 | 11,540 | 31,561 | | |
| Total | 16,551 | 47,223 | 4,952 | 14,976 |

### 3.2 Annotation analysis

3.2.1 Distribution of object categories and their quantities

The efficacy of object detection model is highly influenced by the distribution of object classes in training datasets. The histogram of Figure 2 reveals significant class imbalances of the datasets that could potentially skew model performance. A significant preponderance of 'person' totalling 15,576 sample across image files in trainval dataset, brings to light a disproportionate representation among categories. Contrastingly, certain classes such as 'cow', 'sheep', 'bus', and 'boat' have considerably fewer data.

Class imbalance is a significant factor that may influence the performance of machine learning models. A tendency for models to favour the recognition of more frequently occurring classes is often observed, potentially leading to the neglect of less common categories. Such bias could result in suboptimal model generalization, particularly underrepresented in the training data.

3.2.2 Bounding box analysis

The statistics of bounding box characteristics within the dataset reveal significant diversity in dimensions, size, aspect ratio, and positioning, as shown in Table 2.

**Table 2**. Summary of the statistics of the bounding box in VOC2007 and VOC2012 trainval datasets.

| | Width | Height | Size | Aspect Ratio | Central x | Central y |
|---|---|---|---|---|---|---|
| **Count** | 47,223 | 47,223 | 47,223 | 47,223 | 47,223 | 47,223 |
| **Mean** | 154 | 157 | 34,231 | 1.09 | 239 | 206 |
| **SD** | 133 | 113 | 43,611 | 0.84 | 114 | 75 |
| **Min** | 2 | 4 | 14 | 0.07 | 5 | 6 |
| **25%** | 49 | 62 | 3,128 | 0.55 | 158 | 162 |
| **50%** | 108 | 131 | 14,322 | 0.87 | 240 | 202 |
| **75%** | 227 | 235 | 49,989 | 1.37 | 314 | 251 |
| **Max** | 499 | 499 | 249,001 | 20.63 | 498 | 495 |

### *Multi-scale objects*

Examining the distribution of bounding box sizes and aspect ratios can reveal the variability in object scales, the average dimensions of bounding boxes stand at 154 pixels in width and 157 pixels in height, with standard deviations of 133 and 113, highlighting considerable size variability. This is further evidenced by the wide range of dimensions, stretching from as narrow as 2 pixels to as wide as 499 pixels in both width and height. The data illustrates a substantial variance in the dimensions of bounding boxes, highlighting the presence of both exceptionally large and notably small objects as depicted in Figure 3.

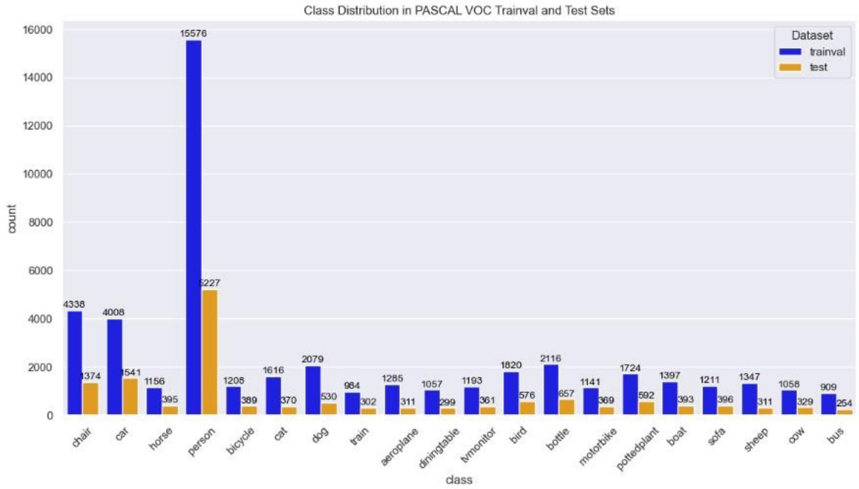


**Figure 2**. Summary of the objects in VOC2007 and VOC2012 trainval and test datasets.

### *Small object*

Despite advances in object detection via deep neural networks, Small Object Detection (SOD) remains a challenge due to the poor visual appearance and noisy representation caused by the intrinsic structure of small targets [21]. The small object is defined as median of relative areas (the ratio of the bounding box area over the image area) of all the object stances in the same class is between 0.08% to 0.58% [22]. In the trainval dataset, it is discovered that there are 5,554 small objects in 2,224 images from 47,223 objects in total.

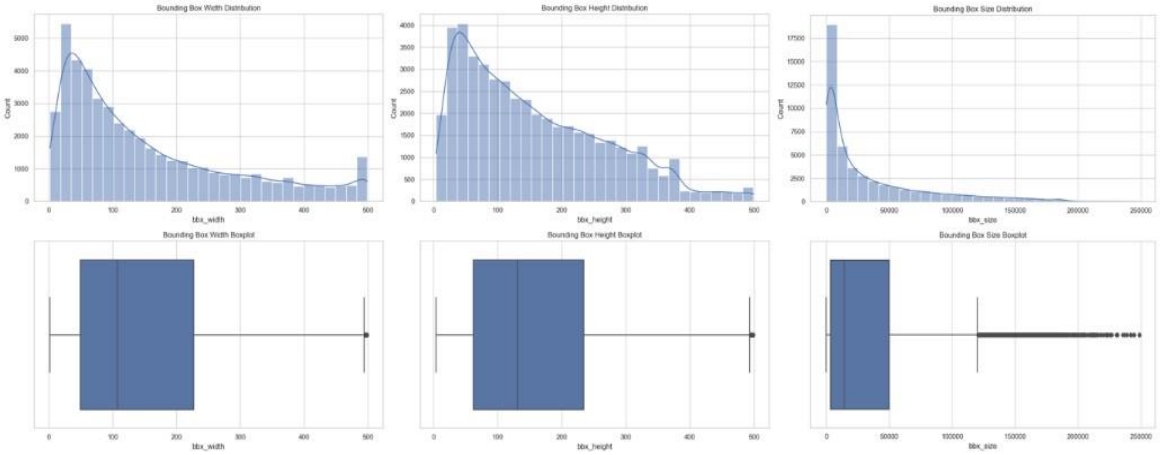


**Figure 3.** Summary of the scale of the bounding box in VOC2007 and VOC2012 trainval datasets.

### 3.3 Image analysis

3.3.1 High-order semantic activity area analysis

As shown from Figure 4, analysing high-order semantic activity areas, it reveals an uneven distribution within the feature map. Certain regions exhibit higher concentrations of semantic information compared to others, indicating disparities in the representation of complex concepts or objects. This uneven distribution suggests potential challenges in accurately capturing and leveraging high-level semantic features across the entirety of the feature map, which may impact the model's ability to effectively recognize and classify objects with varying degrees of complexity and context.

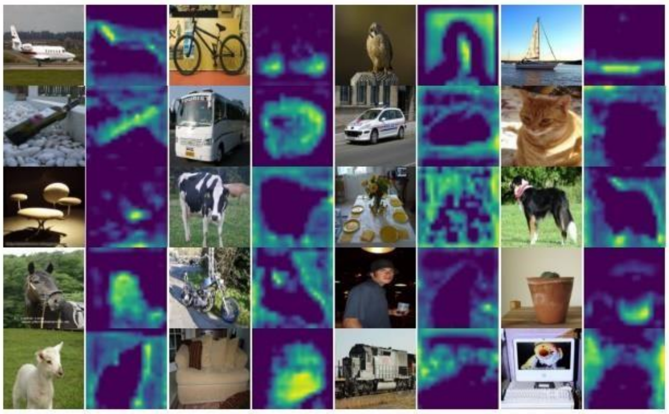


**Figure 4.** Distribution of high-order semantic activity areas

### 3.4 Data preprocessing

3.4.1 Class imbalance problem

According to Oksuz et al., data imbalance problem in object detection can be separated into 4 aspects: class, scale, spatial and objective.[23] Regarding class imbalance, Foreground-Foreground Class Imbalance (FFCI) is inevitable as the sample and detected objects could be obtained from countless circumstances. In Figure 5, it shows a classic case of FFCI, particularly for the person object class which possesses roughly 33% for the whole dataset.

To tackle the problem of FFCI, in accordance with Crasto, N. [24], there are three feasible methods:

1) sampling 2) loss reweighing 3) data augmentation

It is said that sampling and loss reweighing can be benefit, and the augmentation technique, especially for mosaic and mixup, can saliently enhance the YOLO model’s mean Average Precision, by increasing the variability and complexity. Thus, we initially decided to implement sampling and data augmentation during the whole process.

3.4.2 Entropy-based Sampling for Solving Class Imbalance

To begin with, an indicator to quantify the performance of sampling would be vital. Based on the works from Li et al. [25], using entropy to evaluate the degree of class imbalance can be viable. Underlying formular below:

$$H(X) = E[I(X)] = E[-\log_a(P(X))] = -\sum_{i=1}^{n} p_i \log_a p_i \qquad (1)$$

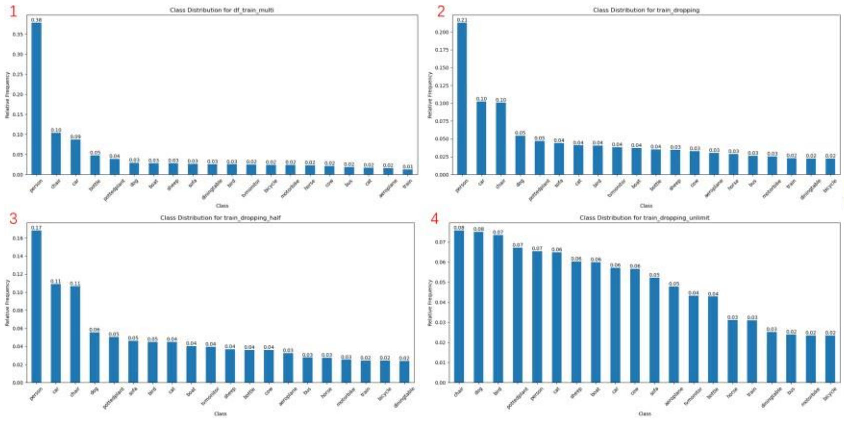
Firstly, we implement a Stratified Shuffle Splitter for splitting the dataset. However, the entropy does not improve. Since under-sampling is a plausible method for FFCI, we decided to conduct an under-sampling on the whole dataset based on the following rules:

1. Whether the sample image contains single or multiple objects.
2. Whether the sample contains the highest ratio object except the lowest ratio object, if yes, mark it as a potential target to drop.
3. Setting the drop step and threshold, start looping
4. Stop when it reaches the threshold or approaches an anticipated entropy.

Here we conduct three experiments for testing the performance of the program, which are:

1. Eliminate by proportion (entropy is 2.743)
2. Eliminate half of the samples (entropy is 2.802)
3. Eliminate until conditions are not met (entropy is 2.927)

In Figure 6, under-sampling based on the above rule can reduce the class imbalance. To evaluate the improvement of entropy, we calculate the relative entropy increase by dividing the maximum value of the entropy when it reaches its limit (that is, all classes are balance in volumes with its entropy of 2.996). Comparison shows in the Figure 5.



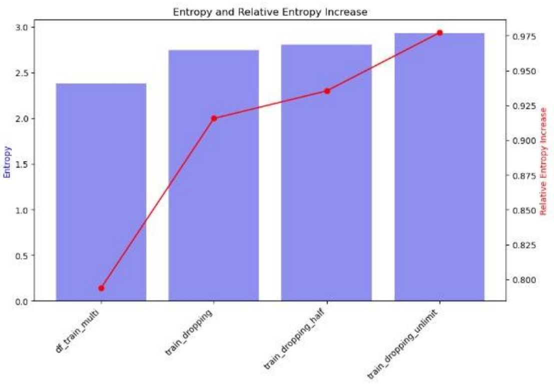


**Figure 5.** Class Distribution for experiment **Figure 6.** Experiment Result of Under-sampling

Hence, we choose the third output with its best entropy, and the sample size is 2703 (7286 objects) with its entropy of 2.927. Beside this under-sampling method, we also conduct sampling by the bounding box number and NIQE (Natural Image Quality Evaluator) score [26] for obtaining the training dataset, and finally we get an original training dataset with 2236 samples (4661 objects) and an entropy of 2.906, which means that the class are relatively uniform.

### 3.5 Discussion

In the data analysis, several key issues have emerged from our dataset evaluation. Firstly, we observed an imbalance in the distribution of image categories, indicating a disparity in the quantity of images representing each class. Secondly, there was inconsistency in the scale and size of the identified objects within the images. Lastly, we noted non-uniformity in the distribution of high-level semantic features among the recognized objects.

To address the first issue of category distribution imbalance, we implemented corresponding preprocessing methods tailored to rebalance the dataset. These preprocessing techniques aimed to mitigate the impact of skewed class distributions, thus improving the model's ability to generalize across different categories.

Regarding the latter two challenges, we proposed a solution leveraging the YOLOv8 architecture. Specifically, we introduced two distinct attention mechanisms to tackle the issues of varying object scales and non-uniform distribution of high-level semantic features. These attention mechanisms were designed to selectively focus on relevant regions within the images, thereby enhancing the model's capability to accurately identify objects of different sizes and semantic complexities.

## 4 Model implementation

### 4.1 Baseline model: Yolov8

YOLOv8, a cutting-edge iteration of the YOLO object detection model developed by Ultralytics, integrates a robust architecture comprising four essential components: the backbone, neck, head, and loss function. The backbone, built upon CSPDarknet53[27] with C2f modules, efficiently extracts hierarchical features from input images, enhancing information flow while minimizing computational costs with the inclusion of the SPPF module. The neck layer combines FPN and PAN techniques from YOLOv5, optimizing feature fusion across multiple scales. Meanwhile, the head module excels in detecting objects of various sizes by transforming feature maps into precise predictions for object location, class, and bounding box dimensions within images.

### 4.2 The proposed YOLOv8-NSK Algorithm

The official YOLOv8 code offers five model versions: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, each with different network widths and depths. This paper selects YOLOv8n as the baseline model for object detection and enhances it further. The proposed model prioritizes fast detection speed, high stability, enhancing the recognition capability of multi-scale and small-scale features.

The Selective kernel networks and Normalization-based attention module are integrated into C2f module and neck layer respectively as shown in Figure 8.

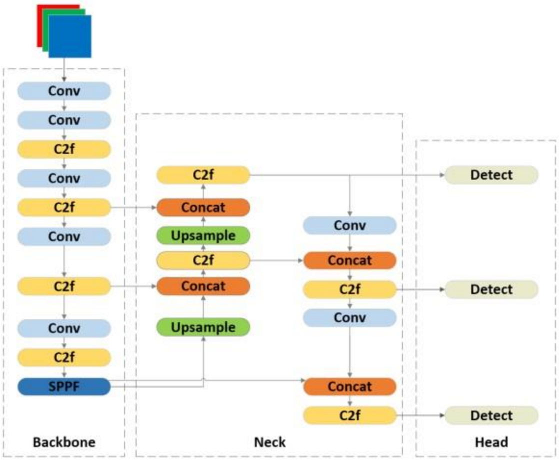


**Figure 7.** Standard YOLOv8 model structure

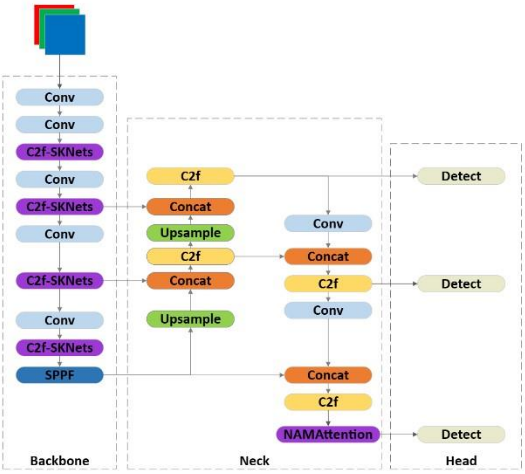


**Figure 8**. Improved YOLOv8-NSK model structure

4.2.1 Efficient Attention Mechanism

4.2.1.1 Selective Kernel Networks

Selective Kernel Networks (SKNets) introduce a novel concept called the Selective Kernel unit. This unit allows neurons within Convolutional Neural Networks (CNNs) to dynamically adjust their receptive field size based on the input information. It achieves this by integrating multiple branches with different kernel sizes. A softmax attention mechanism, guided by information from each branch, determines the contribution of each branch to the final output. This dynamic attention distribution leads to varying effective receptive field sizes at the fusion layer. By stacking multiple SK units, SKNets create a deep architecture that empowers neurons to effectively capture objects of various scales. This demonstrates their capability to adaptively adjust their receptive field based on the input, improving the network's ability to handle objects of different sizes.

4.2.1.2 Normalization-based attention module

Identifying less salient features is crucial for model compression. However, this aspect has not been thoroughly explored in revolutionary attention mechanisms. Normalization-based Attention Module (NAM) suppresses less salient weights. It applies weight sparsity penalties to attention modules, thereby enhancing computational efficiency while maintaining similar performance levels. Within the channel attention submodule, a scaling factor sourced from batch normalization is employed. This factor gauges the variance of channels, serving as an indicator of their importance. To suppress less salient weights, a regularization term is added into the loss function. NAM can effectively aid in identifying and emphasizing high-level semantic features while suppressing unimportant or noisy features to improve model efficiency and performance.

### 4.3 Reproduced yolov5 model

In this study, we reproduce the yolov5 model using Pytorch. The process involves model construction, training, and evaluation.

The dataset employs Mosaic for data augmentation, combining four training images to enrich object backgrounds efficiently. The model architecture as shown in Figure 9, utilize the CSPDarknet network for feature extraction, segmented processing, and reducing parameters and computational complexity. It consists of a residual network, CSPnet, Focus, and SiLU activation function. Feature Pyramid Network (FPN) extracts three feature layers, with bottom layer performing 1x1 convolutional up sampling and top layer performing 3x3 convolutional down sampling. The resulting enhanced features are passed to the HEAD section for predictions.

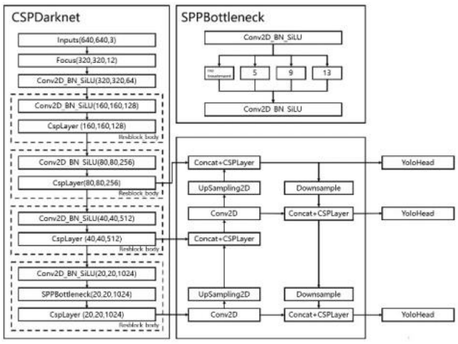


**Figure 9.** Structure of reproduced YOLOV5

# 5 Experiment and result

## 5.1 Experiment setup

### 5.1.1 Datasets

The proposed YOLOv8-NSK neural network was evaluated using the modified PASCAL VOC dataset, which integrates both PASCAL VOC2007 and PASCAL VOC2012. This dataset encompasses 20 target classes and was partitioned into training, validation, and test sets with a 7:2:1 ratio after data preprocessing.

### 5.1.2 Model training and evaluation

The training process of the improved YOLOv8 model and other baseline models is based on RTX 4090(24G), and the dataset partitioning ratio used by all models is consistent. The model is trained with an image size of 640x640 for 100 epochs using a batch size of 64. The learning rate is set to 0.01 with a momentum of 0.937 and a weight decay of 0.0005. The optimizer employed is stochastic gradient descent (SGD). The baseline model loss consists of VFL_loss, CIOU_loss, and DFL.

### 5.1.3 Performance metrics

To comprehensively evaluate the effectiveness of our proposed method, we utilize a diverse set of metrics encompassing precision, recall, mean average precision (mAP), parameter count, and model size. Precision quantifies the proportion of true positives among all predicted positive samples. Mathematically, it is expressed as: $Precision{=}TP/(TP{+}FP)$. Recall is calculated as follows: Recall $= TP/(TP+FN).$ F1 score is defined as: F1 $= 2 \times Precision \times Recall/(Precision + Recall)$. The formula for calculating mAP is as follows: AP $= \int_0^1 P(R)dR$, mAP $= (\sum_{i=1}^{N} AP_i)/N$.

## 5.2 Result analysis

### 5.2.1 Comparison experiment result study

#### 5.2.1.1 Comparison of classical YOLO series methods

To gain insights into the performance of various YOLO series methods, this study first conducts a comprehensive comparison based on the VOC dataset. The comparison framework evaluates key metrics such as precision, mAP50 score, and model size to gauge the effectiveness of each method. The specific modelling results are compared as shown in Table 3. Upon analysis, it is evident that YOLOV8 emerges as a standout performer. With a precision score of 0.718 and a corresponding mAP50 score of 0.715, YOLOV8 exhibits superior accuracy in positive class predictions. Interestingly, despite ranking second in model size with 3 million parameters, YOLOV8's performance surpasses that of other models. This underscores its efficiency in balancing precision and model complexity.

**Table 3.** Comparison of modeling methods

| Models | Precision | Recall | F1-Score | mAP50 | FPS | Params |
|---|---|---|---|---|---|---|
| YOLOV3 | 0.675 | 0.587 | 0.628 | 0.633 | 54.09 | 4.06M |
| YOLOV5 | 0.663 | 0.575 | 0.616 | 0.624 | 46.32 | 2.51M |
| YOLOV6 | 0.67 | 0.58 | 0.622 | 0.624 | 46.27 | 4.24M |
| YOLOV8 | 0.718 | 0.761 | 0.739 | 0.715 | 48.82 | 3M |
| YOLOV8-NSK(ours) | 0.736 | 0.663 | 0.698 | 0.727 | 29.05 | 5.3M |

#### 5.2.1.2 Comparing different attention mechanisms

To evaluate the effectiveness of the Selective Kernel Networks (SKNets), this study incorporated different attention modules, namely Normalization-based Attention Module (NAM), Spatial Group-wise Enhance (SGE) and effective SE(eSE) module. And all modules are inserted into the neck layer of the YOLOv8 network architecture. As observed from Table 4, SKNets outperforms other attention mechanisms in terms of F1-Score and mAP50. Specifically, SKNets achieves the highest F1Score of 0.692 and mAP50 of 0.718, indicating its superior ability to balance precision and recall and effectively capture object instances with high accuracy. Additionally, while SKNets exhibits slightly lower precision compared to eSE, its higher recall compensates for this difference, resulting in a stronger overall performance, demonstrating its higher ability of capturing muti-scale object.

#### 5.2.1.3 Attention mechanisms at various positions

In order to further demonstrate the effectiveness of the SKNets attention mechanism and compare the performance difference of attention mechanisms at various positions, this study attempted to integrate four attention modules into three different positions of Yolov8 baseline, namely the backbone (excludes C2f), neck, and C2f module. Experimental results as shown in Figure10, reveals that SKNet exhibits the best performance when embedded within the C2f module, while its impact on model performance is less pronounced in other parts.

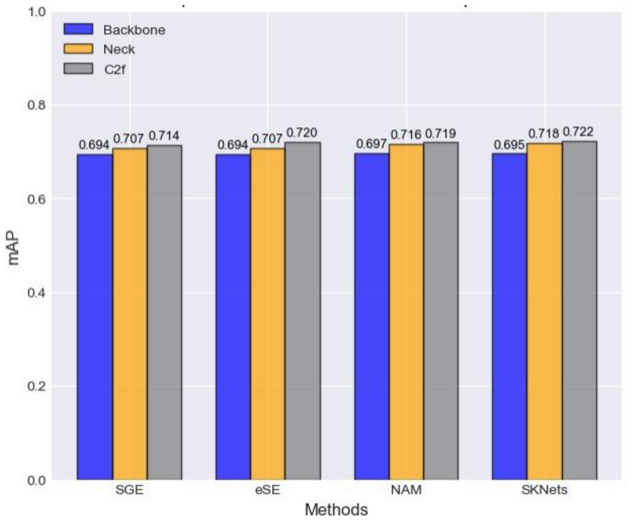


**Figure 10.** Comparisons of mAP in Different Positions

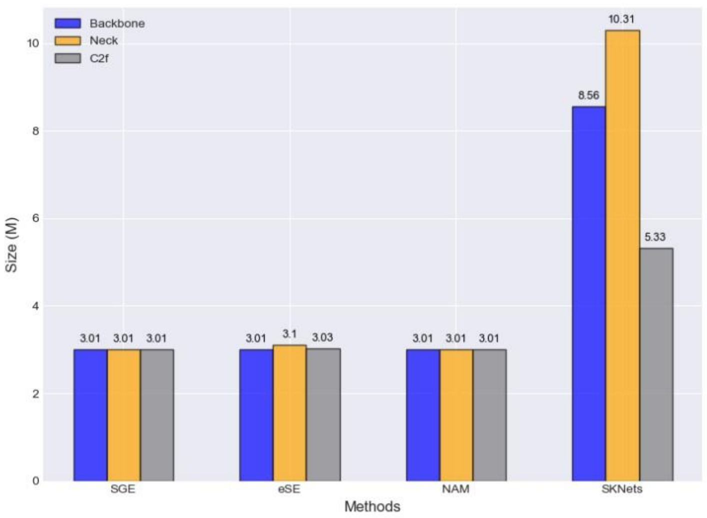


**Figure 11.** Comparisons of Size in Different Positions

5.2.2 Ablation experiment result study

The ablation experiment was carried out to analyse how changes in network architecture impact network performance. This study utilized VOC dataset and investigated four variations: YOLOv8, YOLOv8-NAM, YOLOv8-SKNets and YOLOv8NAM-SKNets. The results of these experiments are detailed in Table 5.

Table 5 presents the validation outcomes using the PASCAL VOC dataset. Substituting the YOLOv8 C2f module with C2fSKNets led to a increase in model accuracy by 0.7% and adding NAM attention module into neck layer of YOLOv8 increase the mAP by 0.1%. Introducing the NAM module in conjunction with C2f-SKNets notably enhanced model accuracy by 1.2%.

**Table 4.** Comparison of Attention mechanism modules

| YOLOV8+Modules | Precision | Recall | F1-Score | mAP50 | FPS | Model-Size |
|---|---|---|---|---|---|---|
| YOLOV8+SGE | 0.729 | 0.636 | 0.679 | 0.707 | 46.19 | 3.01M |
| VOLOV8+eSE | 0.737 | 0.64 | 0.685 | 0.707 | 49.56 | 3.10M |
| YOLOV8NAM | 0.724 | 0.645 | 0.682 | 0.716 | 50.01 | 3.01M |
| YOLOV8+SKNets | 0.728 | 0.66 | 0.692 | 0.718 | 43.84 | 10.31M |

### 5.3 Discussion

In this study, we focused on enhancing the YOLOv8 model, a representative of the YOLO series, based on the following considerations. The selection was mainly narrowed down to one-stage models due to their generally superior performance compared to two-stage models. The YOLO series, known for its real-time capability and accuracy, emerged as the primary focus. At the meantime, through comprehensive performance comparisons among YOLOv3, YOLOv5, YOLOv6, and YOLOv8 as presented in Table 3, YOLOv8 consistently outperformed other models across various metrics including Recall and mAP. This superiority further solidified the choice of YOLOv8 as the baseline model for the research. Furthermore, when comparing YOLOv8 with the transformer-based DETR model, it was found that YOLOv8 offers a more lightweight network structure with approximately 3 million parameters, whereas DETR comprises around 30 million parameters. This stark difference makes DETR significantly challenging to deploy and train due to its large parameter size and prolonged training cycles. Lastly, despite DETR's substantial parameter count, its performance improvement over YOLOv8 within the same order of magnitude was not significant. Consequently, YOLOv8 was deemed a more pragmatic choice for the research objectives.

From Table 4, results reveal that four different attention mechanisms were compared regarding their impact on the YOLOv8 model when integrated into its neck portion. The findings show varied performance changes upon the inclusion of these attention mechanisms. The introduction of Selective Kernel (SK) convolution, particularly in the context of SKNets, presents a novel approach to adaptively adjusting receptive field (RF) sizes within the YOLOv8 model. SK convolution operates through three key operators: Split, Fuse, and Select, facilitating the integration of multiple kernels with different sizes. By leveraging these operators, SKNets effectively capture essential features across various scales, which is crucial for datasets like VOC where objects exhibit significant scale variations.

The integration of attention mechanisms within the C2f structure of YOLOv8 led to substantial performance gains as shown in Figure 10, surpassing those achieved by incorporating attention mechanisms into other convolutional layers or the output sections of the neck structure. This improvement is attributed to three key factors: Firstly, the C2f module's robust feature extraction capability enables effective extraction of local and global features. Attention mechanisms help focus on pertinent image parts, enhancing feature extraction. Secondly, as an intermediate layer, the C2f module operates on partially extracted features, benefiting from attention mechanisms to filter out noise and redundancy. Lastly, the C2f module in the backbone combines features extracted from shallow layers with those from the neck, forming a fused representation. By introducing attention mechanisms at this juncture, the model can effectively highlight the primary features within the fused representation, thereby amplifying model performance.

**Table 5.** Performance results of different models on VOC07+12

| No. | NAM | SKNets | mAP(%) |
|---|---|---|---|
| 1 | | | 71.5 |
| 2 | √ | | 71.6 |
| 3 | | √ | 72.2 |
| 4 | √ | √ | 72.7 |

In ablation experiment as shown in Table 5 we conducted ablation experiments on YOLOv8 by integrating the NAM (Normalization-based Attention Module) and SKNets (Selective Kernel Networks) attention mechanisms into the model architecture. Firstly, we incorporated the NAM module into the last layer of the neck structure, utilizing the feature maps of maximum scale containing high-level semantic features as input. The NAM module effectively suppressed noise features, identified, and preserved weights of high-level semantic features, thereby significantly enhancing recognition accuracy. Secondly, SKNets inherently possess cross-scale object recognition capabilities for extracting primary features. Integration of the SKNets with the C2f module amplified the multi-scale object recognition capabilities of the C2f module itself, resulting in performance improvements. Combining these attention mechanisms resulted in a notable 1.2% enhancement in model recognition accuracy.

## 6 Conclusion and future work

In the age of deep learning, it's crucial to assess the ethical implications of deploying machine learning methods. While advancements like the improved YOLO model offer better accuracy, they also raise concerns about privacy, bias, and accountability. Large datasets and powerful algorithms can perpetuate biases, leading to unfair outcomes. Deploying AI in areas like law enforcement and healthcare requires careful consideration of individual rights and societal values. As researchers, we must prioritize ethics, transparency, and dialogue with stakeholders to ensure responsible development and deployment of machine learning for society's benefit.

This study identified three main issues within the dataset through thorough analysis: 1) imbalanced class distribution, 2) diverse object scales, and 3) uneven distribution of high-level semantic features. To address the challenges of uneven data category distribution, we proposed corresponding data preprocessing techniques. Additionally, we tackled the issues of inconsistent recognition category scales and uneven distribution of high-order semantic features by introducing SKNets and the NAM attention mechanism. Extensive comparative experiments demonstrated that integrating SKNets into the C2f module yields the most significant advantages, while incorporating NAM into the Neck layer produces the most optimal results. Experimental results further validate that our proposed YOLOv8-NSK model achieves a 1.2% increase in mAP accuracy on the VOC dataset compared to YOLOv8. This research contributes to the advancement of object detection techniques and provides valuable insights for future studies in this field.

Looking ahead, our research will prioritize further advancements in following key areas:

1. Enhance the backbone architecture of YOLOv8 by incorporating the design principles of knowledge distillation or pruning, aiming to reduce model size, parameter count, and computational complexity.
2. Investigate self-supervised learning techniques through the creation of tailored self-supervised tasks. These tasks should enable the model to autonomously acquire valuable features, consequently improving the efficacy of small object detection.

## Reference

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[2] M. Everingham, L. Gool, C. Williams, J. Winn, & A. Zisserman, "The pascal visual object classes (voc) challenge" *International Journal of Computer Vision*, vol. 88, no. 2, p. 303-338, 2009. https://doi.org/10.1007/s11263-009-0275-4-4.

[3] X. Li, W. Wang, X. Hu and J. Yang, "Selective Kernel Networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 510-519, doi: 10.1109/CVPR.2019.00060.

[4] Y. Liu, Z. Shao, Y. Teng, and N. Hoffmann, "Nam: Normalization-based Attention Module," 2021, *arXiv*:2111.12419.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.

[6] C. Shu, X. Ding and C. Fang, "Histogram of the oriented gradient for face recognition," in *Tsinghua Science and Technology*, vol. 16, no. 2, pp. 216-224, April 2011, doi: 10.1016/S1007-0214(11)70032-3.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, Sept. 2010, doi: 10.1109/TPAMI.2009.167.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional Neural Networks," in *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017. doi:10.1145/3065386

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014. doi:10.1109/cvpr.2014.81

[10] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015, doi: 10.1109/TPAMI.2015.2389824.

[11] R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fuet al., "Ssd: single shot multibox detector", in *Computer Vision – ECCV 2016*, p. 21-37, 2016. https://doi.org/10.1007/978-3-319-46448-0_2.

[13] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 1 Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, & S. Zagoruyko, "End-to-end object detection with transformers", in *Computer Vision – ECCV 2020*, p. 213-229, 2020. https://doi.org/10.1007/978-3-030-58452-8_13.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomezet al., "Attention Is All You Need" 2017, *arXiv*:1706.03762.

[16] X. Zhu, "Deformable DETR: Deformable Transformers for End-to-End Object Detection" 2020, *arXiv*:2010.04159.

[17] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 9992-10002, doi: 10.1109/ICCV48922.2021.00986.

[18] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhuet al., "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection" 2022, *arXiv*:2203.03605

[19] Y, Zhao, et al. "DETRs Beat YOLOs on Real-time Object Detection," 2023, *arXiv*:2304.08069.

[20] T.-Y. Lin et al., "Feature Pyramid Networks for Object Detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. doi:10.1109/cvpr.2017.106.

[21] G. Cheng et al., "Towards large-scale small object detection: Survey and benchmarks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, Nov. 2023. doi:10.1109/tpami.2023.3290594.

[22] C. Chen, M. Li, O. Tuzel, & J. Xiao, "R-cnn for small object detection", *Computer Vision – ACCV 2016*, p. 214-230, 2017. doi: 10.1007/978-3-319-54193-8_14.

[23] K. Oksuz, B. C. Cam, S. Kalkan and E. Akbas, "Imbalance Problems in Object Detection: A Review," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3388-3415, 1 Oct. 2021, doi: 10.1109/TPAMI.2020.2981890.

[24] N Crasto, "Class Imbalance in Object Detection: An Experimental Diagnosis and Study of Mitigation Strategies" 2024, *arXiv*:2403.07113.

[25] L. Li, H. He and J. Li, “Entropy-based Sampling Approaches for Multi-Class Imbalanced Problems,” in *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2159-2170, 1 Nov. 2020, doi: 10.1109/TKDE.2019.2913859.

[26] A. Mittal, R. Soundararajan and A. C. Bovik, “Making a “Completely Blind” Image Quality Analyzer,” in *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209-212, March 2013, doi: 10.1109/LSP.2012.2227726.

[27] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018, *arXiv*:1804.02767.