修 士 論 文 概 要 書

Master's Thesis Summary

Date of submission: 07/24/2023 (MM/DD/YYYY)

専攻名 (専門分野) Department	Computer Science and Communications Engineering	氏 名 Name	Sirui Liu	指 導 教 員 Advisor	印 Hiroshi Watanabe Seal
研究指導名 Research Guidance	Research on Audiovisual Information Processing	学籍番号 Student ID number	5121FG58-3		
研究題目 Title	Research on The High-Resolution Image Inputs in Edge AI Systems for Human Detection Using YOLOv7				

1. Introduction

The concern of this thesis is object detection in the context of high-resolution inputs for edge-cloud artificial intelligence systems. The bottlenecks and challenges faced by this topic are high-resolution images or videos leading to increased computational complexity, insufficient edge arithmetic, and insufficient transmission bandwidth. In addition, traditional feature extraction struggles to effectively capture the detailed features of high-resolution objects, leading to a decrease in accuracy. Various solutions have been proposed, such as cutting images for individual detection or using multi-scale processing mechanisms, but each has its limitations. This thesis proposes a pedestrian detection framework that makes the detection speed and inference latency tunable so that the results are both accurate and fast.

2. Related works

2.1 Object detection model-YOLOv7

Although various target detection models have emerged in an endless stream in recent years, YOLOv7 still achieves a new high in detection speed and accuracy. Several trainable bag-of-freebies are designed in the model so that the real-time detector can greatly improve detection accuracy without increasing the inference cost.[1]

For the development of target detection, YOLOv7 also raised two new issues, namely, how to effectively replace the original module with module reparameterization, and how to deal with the assignment of different output layers by the dynamic label assignment strategy. The model proposes "extend" and "compound scaling" methods for real-time detectors, which can use parameters and calculations more efficiently. At the same time, the method proposed in the model can effectively reduce the parameters of real-time detectors by 50% and has faster inference speed and higher detection accuracy.

2.2 Methods of high-resolution input

In high-resolution object detection, research has focused on processing large images. The paper *"You Only Look Twice"* [2] proposes methods to address small object detection and data imbalance. It uses feature maps from different sensory fields to enhance small object detection and employs multiple scales of data for training, along with data augmentation. The method uses sliding window clipping to segment oversized images, which are then processed separately and merged using NMS.

Another paper, *"Flexible High-resolution Object Detection on Edge Devices with Tunable Latency,"* [3] suggests efficient image segmentation to separate regions with dense and sparse objects. Different parametric models are used for detection, assigning large models to computationally demanding regions for accuracy and small models for faster inference and reduced latency. The method generalizes the model using adjustable latency constraints.

2.3. Benchmark and baseline

The Panda dataset is used in this research to provide super-resolution images for training and validation. This is a human-centered video dataset at the gigapixel level, which contains a variety of real-world scenarios. Each image frame in this dataset can reach a size of 26753*15052 pixels, which is much larger than the regular data volume for object detection tasks. It provides 18 scenes and more than 15,974.6 k of bounding box annotations.

The benchmark provides the results of Cascade RCNN, Faster RCNN, and RetinaNet as the baseline, and the experimental results in this thesis will be compared with this baseline to observe its effectiveness.

The evaluation criteria of this research will use the Microsoft Coco metrics to judge the effectiveness of the proposed method.

3. Proposed method

In this approach, a stable and fast human detection model is first selected. The relationship between human size and model performance is evaluated to generate a size vs. precision curve. By choosing a precision threshold (p), the minimum human size (S) required to achieve the desired precision is determined.

Next, the image is divided into patches, and the size range of bounding boxes containing the pixel point is recorded for each patch (calculated by

extracting frames from the video). Heat maps are generated based on the human size range in each patch.

To mark regions containing patches where human sizes range from [0 to 2S), minimum rectangles are utilized while maintaining the original resolution for this type. Similarly, minimum rectangles are used for regions containing patches with human sizes ranging from [2^nS to 2^{n+1}S), but these regions are down-sampled to $1/2^n$ *$1/2^n$ size.

The compression-processed individual sub-graphs are then transmitted to the cloud for inference, and the inference results are sent back to the edge devices. Finally, the inference results are fused, and post-processing steps like NMS are performed to refine the final output.

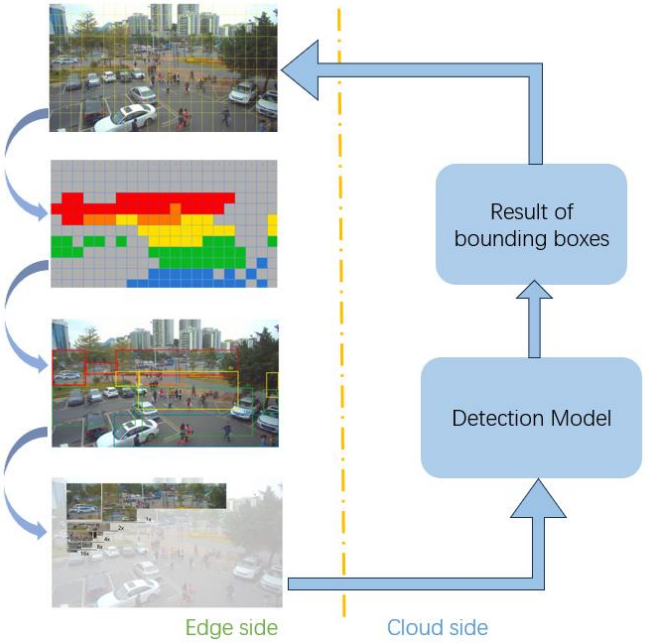

Fig. 1. Overview of the framework

We also provide a detailed description of the processing techniques of each module in the thesis and explain it with an example of a scenario.

4. Experiments

The code in this experiment was written by Python 3.7.13, Pytorch 1.12.0 and executed under the Ubuntu 20.04 operating system. The graphics card used in the study is Nvidia RTX A6000.

In this experiment, the initial threshold of AP is set to 0.4, and according to the fitting curve of human size and AP we tested, we set the minimum value of size at 64 pixels. and the scene image is divided into regions with *S=64* pixels as the standard. And we take one of the 30 frames of the video as historical data, which is used to analyze the scene. We used five scenes in the dataset to do the test and got the data results in Table 2.

Table I Results of Our Proposed Method and The Baseline

	AP	Average Time/s	FPS
Cascade R-CNN	0.30889	20	0.05
Faster R-CNN	0.28442	14.29	0.07
RetinaNet	0.22545	10	0.1
YOLOv7*	0.483	2.91	0.337
Proposed Method	0.412	**0.486**	**2.06**

In the table, YOLOv7* means that the input image was down-sampled by a factor of 2, and then a sliding window with a size of 640*640 pixels was used to do the detection.

It can be seen that our proposed scheme outperforms the three models of baseline in terms of both accuracy and inference time.

5. Conclusion

In summary, our research focuses on a tunable framework for pedestrian detection that is highly efficient in both accuracy and speed.

Since our method divides the original data, it can support parallel computing of multiple small-scale pictures or videos. And it is flexible to explore potential applications with other algorithms or detection models.

In addition, from the perspective of data transmission, the feasibility of computing some of the compressed data images directly on the camera side can be considered, while other images can be efficiently processed in the cloud. Overall, our proposed method in this paper demonstrates its compatibility with various algorithms and its potential for super-resolution image object detection in edge-cloud systems.

6. Reference

[1] C.Y. Wang, A. Bochkovskiy, H. Yuan and M. Liao: “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”, 6 Jul 2022; [https://doi.org/10.48550/arXiv.2207.02696]

[2] Adam Van Etten, “You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery”, 24 May 2018; [https://doi.org/10.48550/arXiv.1805.09512]

[3] S. Jiang, Z. Lin, Y. Li, Y. Shu, Y. Liu: “Flexible high-resolution object detection on edge devices with tunable latency”, Pages 559–572, MobiCom ' 2021.

Research on The High-Resolution Image Inputs in Edge AI Systems for Human Detection Using YOLOv7

A Thesis Submitted to the Department of Computer Science and Communications Engineering,
the Graduate School of Fundamental Science and Engineering of Waseda University
in Partial Fulfillment of the Requirements for the Degree of Master of Engineering
Submission Date: July 24th, 2023.

Sirui LIU

(5121FG58-3)

Advisor: Prof. Hiroshi Watanabe

Research guidance: Research on Audiovisual Information Processing

Acknowledgments

I sincerely thank Professor Hiroshi Watanabe for his valuable guidance during my studies at Waseda University, not only for his expertise but also for the moral support and encouragement he gave me.

I also want to thank my parents, my long-distance boyfriend, Du Huayu, and my friend, Song Jinlan, for their constant support and trust. They have been a source of strength for my spirit.

In addition, I would like to thank Waseda University for providing me with a study environment that allowed me to focus on my research, as well as the guidance of the teachers of each course over the past two years.

I will always remember and miss my four years of study in Japan.

Abstract

The concern of this thesis is object detection in the context of high-resolution inputs for edge-cloud-based artificial intelligence systems. The bottlenecks and challenges faced by this topic are high-resolution images or videos leading to increased computational complexity, insufficient edge arithmetic, and insufficient transmission bandwidth. In addition, traditional feature extraction struggles to effectively capture the detailed features of high-resolution objects, leading to a decrease in accuracy. Various solutions have been proposed, such as cutting images for individual detection or using multi-scale processing mechanisms, but each has its limitations. This thesis proposes a pedestrian detection framework that makes the detection speed and inference latency tunable so that the results are both accurate and fast.

Because the edge camera is fixed, the frame composition is stable, and the uniqueness of pedestrian detection. The proposed method in this thesis reduces the amount of computation and data transmission by segmenting and cropping the image according to the object size and compressing the data accordingly for regions with different-sized objects. The compressed data is transmitted to the cloud for detection, and the detection results are post-processed to output the final detection results.

This detection framework experiments on a super-resolution dataset with the YOLOv7 detection model. By comparing the results with the baseline results, the experiment results show that the proposed method can achieve accurate and efficient object detection in high-resolution scenes while reducing computational requirements and bandwidth costs.

Keywords: Edge-AI, human detection, YOLOv7, high-resolution, computer vision.

List of Contents

List of Figures

Chapter1

Chapter2

Chapter3

List of Tables

Chapter 1 Introduction

1.1 Overview of the Task

1.1.1 Background of Object Detection

Object detection is a fundamental research task in the field of computer vision aimed at automatically identifying and localizing specific objects in images or videos. In the past decades, significant progress has been made in object detection technology, providing powerful solutions for numerous practical applications. Among them, pedestrian detection, as an important task of object detection, has extensive value and research significance.

Pedestrian detection is a key computer vision task with applications in various fields. For example, it plays a crucial role in intelligent surveillance systems, traffic management, autonomous driving, and human-computer interaction. Pedestrian detection also makes significant contributions to public safety, urban planning, crowd management, and social sciences. Accurate pedestrian detection enables us to obtain valuable data on demographics, pedestrian behavioral patterns, urban space utilization, etc., thus providing substantial support for decision-making and resource allocation.

However, pedestrian detection encounters several challenges and difficulties. First, pedestrians exhibit great variations in appearance and pose in different scenes and viewpoints, leading to low robustness of detection algorithms against complex backgrounds, occlusions, and pose variations. Second, pedestrian detection tasks require a balance between real-time and accuracy, especially in high-density crowds and complex environments, requiring algorithms with high detection speed and low miss rates. In addition, pedestrian detection needs to address challenges such as annotation difficulties and unbalanced training samples on large-scale datasets.

To overcome these challenges, researchers have proposed innovative methods and

techniques to improve pedestrian detection performance. Traditional machine learning methods and feature extraction techniques, e.g., manual features, Haar features, have achieved some success. However, in recent years, the rapid development of deep learning technologies has brought breakthroughs in pedestrian detection. The introduction of convolutional neural networks and object detection networks such as Cascade R-CNN, YOLO, Faster R-CNN, and SSD has significantly improved the performance and accuracy of pedestrian detection.

1.1.2 Background of object detection on high-resolution images

In the context of high-resolution inputs, object detection faces significant challenges, mainly in computation and perception.

First, high-resolution images or videos increase computational complexity. As the image resolution increases, the size of the object region and the number of pixels also increases, which requires more computational resources and processing time. Due to the need to process more features and a larger sensory field to capture contextual information, traditional object detection algorithms are usually difficult to meet the real-time requirements in high-resolution scenes.

Second, high-resolution images pose perceptual challenges to object detection. Objects in high-resolution images show higher complexity and richer appearance features. Traditional feature extraction methods are more difficult to effectively capture these detailed features, and if the original input is detected after a simple downsampling process, it will lead to the loss of this detailed information. This leads to a decrease in detection accuracy.

To cope with the challenges posed by high-resolution inputs, researchers have proposed various solutions. For example, cutting the images to detect them separately, or using multi-scale processing mechanisms, such as constructing a feature pyramid in the model. However, each method has its advantages and disadvantages, and this thesis addresses these challenges by proposing a solution that is relatively acceptable

in terms of speed and accuracy.

1.1.3 Background of Edge AI

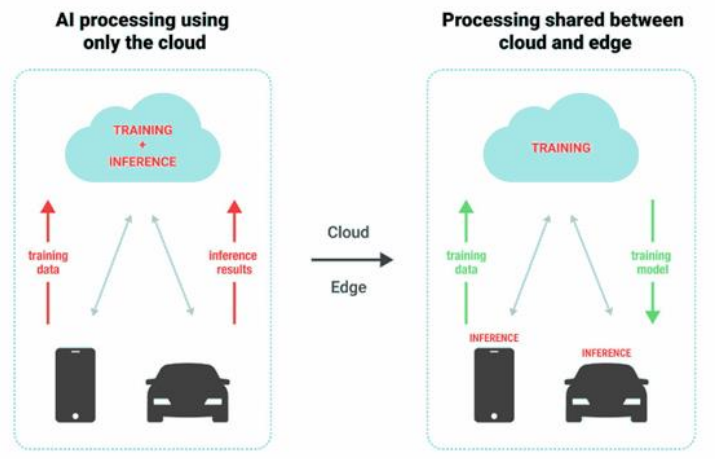

Fig. 1. The difference between Edge AI and Cloud AI

Edge AI refers to deploying AI algorithms and models directly on edge devices such as smartphones, IoT devices, and edge servers, rather than relying exclusively on cloud processing. This paradigm shift has received significant attention and research interest in recent years due to its potential to address the limitations of cloud-centric AI systems.

The main challenges faced by AI systems at the edge, besides having cost issues of needing to transmit large volumes of data and privacy and security concerns, the main technological core lies in latency and bandwidth constraints. Cloud-based AI systems all suffer from high latency issues and require large amounts of bandwidth due to the need to transfer data to remote servers for processing. However, many real-time computing applications, such as video analytics under surveillance cameras and autonomous driving systems in cars, require low latency and high bandwidth. Traditional cloud-centric architectures, on the other hand, while having a large number of computing resources, do so at the cost of latency and data security issues, and therefore cannot meet these requirements. Edge AI aims to overcome these limitations by enabling AI processing and decision-making directly on edge devices,

reducing the need to transmit data to the cloud.

To address these challenges, research in edge AI aims to explore lightweight AI models suitable for resource-constrained edge devices, as well as more efficient edge reasoning frameworks, and edge-cloud coordination mechanisms. What is proposed in this thesis is an inference framework suitable for object detection tasks on edge AI.

1.2 Problem Statement

1.2.1 Bottlenecks in the Edge AI System

Edge devices usually have limited computing power and storage space, so it is difficult to deploy complex deep-learning models and algorithms. This limits the performance and capability of Edge AI systems when dealing with large-scale data and complex tasks. Data exchange with the cloud, on the other hand, requires high bandwidth, so solving this problem requires an edge-cloud framework that contains lightweight models and effectively compresses the amount of data transfer.

Training models on the cloud and then deploying them to edge devices is currently an effective solution for object detection tasks in edge AI systems. In practical application deployment scenarios, reasonable use of contextual information for analysis and computation on the cloud can achieve twice the result with half the effort, while enhancing the generalization ability of the model. The method proposed in this thesis is to reduce the inference delay, reasonably allocate the arithmetic power, and reduce the bandwidth cost by reducing the amount of data transmission between the edge and the cloud.

1.2.2 Characteristics of Pedestrian Detection

Most pedestrian detection in real applications is captured by fixed cameras, such as

street surveillance and car cameras. It is characterized by a more stable overall composition of the image, and the position of the pedestrians appearing in the image is traceable.

And because the person is the object of object detection, the shape and actual size of the detection frame are relatively stable. Since the camera is fixed and stationary, people close to the camera have large sizes and people far from the camera have small sizes in most images. This thesis will use this as one of the theoretical bases for the construction of the frame.

1.3 Thesis Outline

Chapter 1: This Chapter introduces the object detection task background and its application to edge AI systems. We also compare the difference between current cloud computing and edge AI, explain the advantages of edge AI, and explain the purpose of designing this framework in this thesis. The theoretical underpinnings of the approach proposed in this thesis are also explained from two perspectives: latency and bandwidth issues and task characterization in real-world applications.

Chapter 2: We introduce the more advanced and efficient object detection algorithm and discuss its detection effectiveness, i.e., the model of YOLOv7 used in the experiments. In addition, we present the relevant dataset used for training and validation in the experiments, the realizations of other widely used object detection models on this dataset, and the evaluation metrics.

Chapter 3: Through a detailed introduction, we show the framework of the proposed method. The overall structure of this Edge-Cloud-based object detection framework and the purpose of the design of each module in it are presented. In addition, we also provide an expanded description of the processing of each of these modules.

Chapter 4: We briefly describe the experimental setting of the proposed model.

Experiments based on our proposed model were conducted on the dataset and the resultant data were analyzed. And the performance with the existing baseline on this dataset is compared and discussed in terms of two metrics: accuracy and detection speed.

Chapter 5: We summarize the strengths and weaknesses of this thesis and analyze the possibility and acceptability of its deployment in real applications. Finally, the framework's future improvement directions are discussed.

Chapter 2 Related Works

2.1 Previous methods

2.1.1 Object Detection Model

The detection framework proposed in this thesis requires a network with both accuracy and inference speed as a base model. We chose yolov7 for further experiments. YOLOv7 is a real-time object detection model published by the official YOLO team. The research's background and motivation stem from the significance of real-time object detection in computer vision. Devices running real-time detectors are often equipped with mobile CPUs, GPUs, and NPUs. The recent focus on edge devices aims to optimize vanilla convolution, depth-wise convolution, or MLP operations. YOLOv7's proposed real-time object detector better supports both edge mobile GPUs and high computing power GPUs, aligning with the current trend of adapting detectors to edge devices.

This model addresses several challenges, including model structure reparameterization and dynamic label assignment in models with multiple output layers. A new label assignment method, the "coarse-to-fine guided label assignment strategy," is introduced to address dynamic object assignment problems. Three main innovations presented in the method are the design of a trainable "bag-of-freebies," solving the problems of module reparameterization and dynamic label assignment in object detection, and the introduction of the extend and compound scaling methods for real-time detectors. These methods efficiently utilize parameters and computations, resulting in 50% parameter reduction, faster inference speed, and higher detection accuracy.

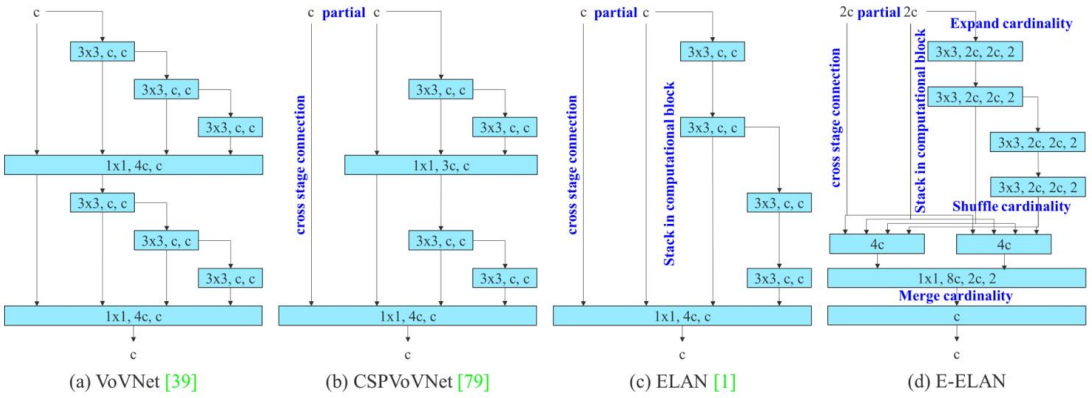

Fig. 2. Extended efficient layer aggregation networks.

In this Fig. 2 (a) VoVNet is an existing backbone for implementing object detection, and CSPVoVNet in (b) is designed based on VoVNet that analyzes gradient paths to enable the learning of weights in different layers to more diverse features. This gradient analysis method enables inference with higher speed and more accurate results. And in (c) ELAN considers how to design an efficient network, and the proposed method for this is to control the shortest and longest gradient path so that deeper networks can learn and converge effectively.

Extended-ELAN based on ELAN is used in this method. The stacking of computing blocks in ELAN reaches a stable state. If more computing blocks are stacked infinitely, this stable state may be destroyed, and the parameter utilization will be reduced. E-ELAN proposed a method that uses expand, shuffle, and merge cardinality to achieve the ability to continuously enhance the network's learning ability without destroying the original gradient path.

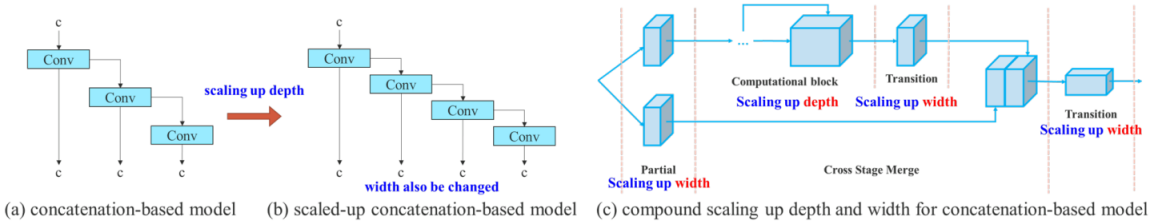

Fig. 3. Model scaling for concatenation-based models

In order to meet the needs of different inference speeds, it is necessary to adjust

some properties of the model and generate models of different scales. The model analyzes the influence of convolution and group convolution on the number of parameters and computation and designs the corresponding model scaling method. For concatenate-based architectures, when the execution depth is enlarged or reduced, the computational blocks of the translation layer will decrease or increase, so different scaling factors cannot be analyzed separately. The compound model scaling method is proposed in the paper. When scaling the depth factor of a computational block, the change of the output channel of the block shoud also be calculated. The transition layer will then be scaled by an equally varying width factor. This preserves the properties of the model as it was originally designed and maintains the optimal structure.

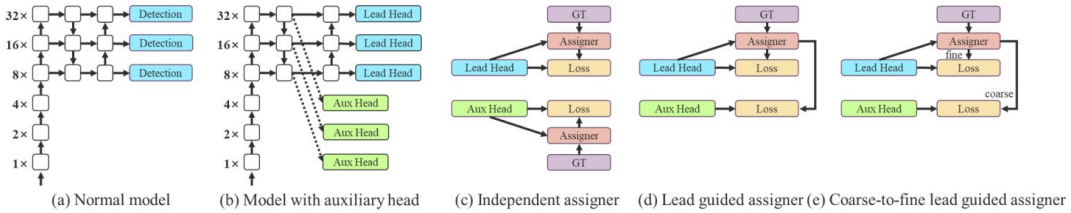

Fig. 4. Coarse for auxiliary and fine for lead head label assigner

Regarding the "Trainable bag-of-freebies," the method introduces "Planned re-parameterized convolution" and "label assignment" as the main technical points. The convolutional structure is based on RepConv, with RepConvN replacing identity connections. Deep supervision is added to enhance training, and the label assignment method employs soft labels from the lead head to guide both the auxiliary head and lead head learning. The Coarse-to-fine lead head guided label assigner generates thick and thin labels to optimize the auxiliary head's recall rate, dynamically adjusting the importance of fine and coarse labels during the learning process.

Yolov7 model has highly improved speed and accuracy compared to other existing object detection models. Its performance surpasses all other detectors within the FPS range of 5 to 160, achieving the highest accuracy of 56.8% AP among real-time object detectors operating at 30 FPS or higher, which is suitable for the experiments in our

method.

2.1.2 Edge AI system

The layered structure of the most common edge computing system is mainly composed of Endpoint, Near edge, Far edge, Cloud, and Enterprise. Among them, the near edge refers to non-standard servers or devices, and the far edge refers to intermediate nodes with strong computing power, such as the cascade server room of cloud service providers or the operator's server room, and so on. The cloud is the proprietary cloud service or shared cloud, which has a large number of centralized computing resources, and these resources are reasonably allocated and centrally managed.

Computing in the cloud and at the edge are not substitutes for each other but rather complement each other to tune for better fulfillment of goals, such as object detection tasks using artificial intelligence. Input data is usually obtained by devices at the edge, and after transferring a large amount of training data to the cloud, the cloud will train neural network models against this data. Heavy computational tasks and storage of model data are deployed in the cloud. After the model training is completed, in practical applications, the edge end sends the acquired real-time data to the cloud, and the object detection model in the cloud inputs it into the network and then sends the inference results to the edge device. Therefore, how to process the real-time data while ensuring accuracy so that the amount of data sent can be compressed, and how to reduce the amount of computation in the cloud is a key issue in this task.

2.1.3 Previous method in high resolution

There has also been research in the direction of processing large-size images in object detection tasks. This thesis also refers to some of these methods. In the paper *You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery*, several processing methods are proposed to overcome the problems of the object

being too small in the image and the imbalance of data volume. It incorporates feature maps of different sensory fields in its model to enhance the detection of small objects. In addition, it employs different scales of data for training as well as data augmentation to increase the percentage of foreground objects in the image. In terms of data, sliding window clipping is used in this method to segment the oversized resolution images, which are processed separately and then NMS processing is used to fuse the detection results of all the sub-images.

Another referenced paper on the task of object detection on high-resolution images is called *Flexible High-resolution Object Detection on Edge Devices with Tunable Latency*. This paper proposes an efficient segmentation of the image so that regions requiring dense and sparse objects are separated. and sparse regions are separated. Then different parametric models are used for detection. Large models are assigned to regions that require more computing power to ensure accuracy, and small models are used to accelerate inference and reduce system latency. Moreover, the method can generalize the model by using the set latency as an adjustable constraint.

2.2 Benchmark and Dataset

In this paper panda dataset is used as high-resolution input. The images and videos in this dataset are captured by fixed cameras on the streets. This is a human-centered video dataset at the gigapixel level which can be used for large-scale, long-term, and multi-object visual analysis. It contains a variety of real-world scenarios, where pedestrians vary even more than a hundred times in scale in the images, posing a huge challenge to the detection task.

Data-wise, each image frame in this dataset can reach a size of 26753*15052 pixels, which is much larger than the regular data volume for object detection tasks. It provides 18 scenes and more than 15,974.6 k of bounding box annotations.

In terms of benchmark format, the panda dataset uses the bounding box annotation format of Microsoft Coco format, which is the most common format for object detection tasks. Therefore, the coco evaluation criteria were also used in evaluating the experimental results. The benchmark provides the results of Cascade RCNN, Faster RCNN, and RetinaNet as the baseline, which are widely used detectors, and the experimental results in this thesis will be compared with this baseline to observe its effectiveness.

2.3 Evaluation Metrics

For the object detection task, the most important thing is that the network model is fast and accurate. A commonly used evaluation metric is generally mAP, which refers to mean average precision mean, accuracy evaluation. For speed, it is evaluated using FPS, which is the number of images processed per second or the time required to process each image.

The metric mAP needs to be calculated from several different parameters. Firstly, the detection frame of each inference result is compared with the annotation frame of the original ground truth to calculate the IOU value. IOU value is calculated as the ratio of the intersection and concatenation of the detection frame and the annotation frame.

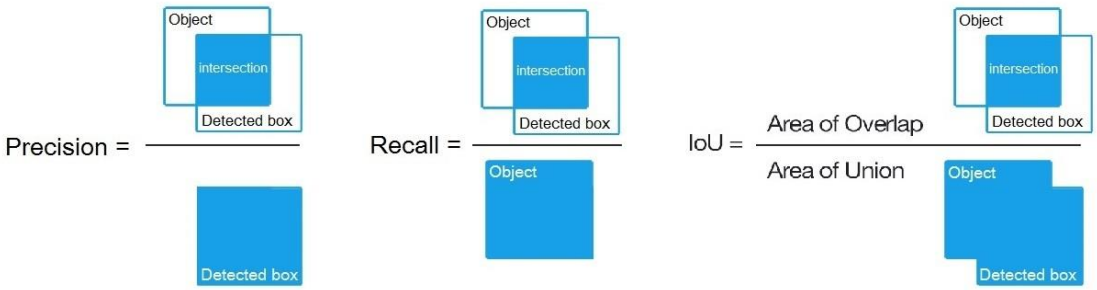

Fig. 5. Formula of Precision, Recall, and IoU

The IOU value is calculated as the ratio of the intersection and concatenation of the

detected frame and the annotation frame, where the detected frame with the IOU value greater than the set threshold is determined as TP, the detected frame with the IOU value less than the set threshold is determined as FP and the non-detected object is determined as FN.

Under this criterion, Precision is calculated as the ratio of TP and TP + FP; Recall is calculated as the ratio of TP and TP + FN, and AP refers to the area under the curve of the relationship between the two values of Precision and Recall, the higher the value of AP, the better the detection effect of the model.

In the Microsoft coco series of evaluation metrics, mAP is the average AP value for each detection category calculated with multiple IOU thresholds from 0.5 to 0.95 in steps of 0.05. In addition to this, there are different AP types differentiated by object size, small, medium, and large. These metrics will be used in experiments and data analysis to evaluate the effectiveness of the model.

Chapter 3 Proposed Method

3.1 Main Idea

In considering the edge-cloud system for object detection incorporates a unique feature of the human detection task. That is, the shape and actual size of the human box is relatively stable. Since the camera is stationary, in most images, people close to the camera are large and people far from the camera are small. Meanwhile, the relationship between the percentage of the object in the image and the detection accuracy in object detection is well documented; the smaller the image, the more likely the feature map will lose information, and the less accurate the detection results will be.

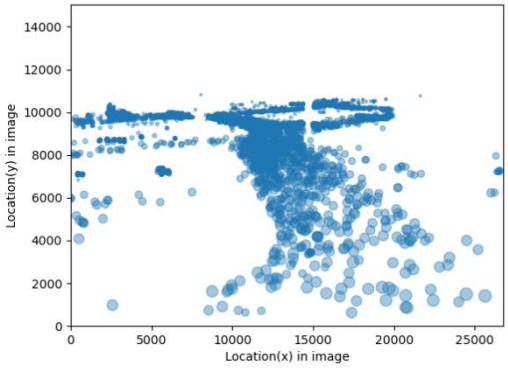

Fig. 6. Distribution of different-size objects in the example scene

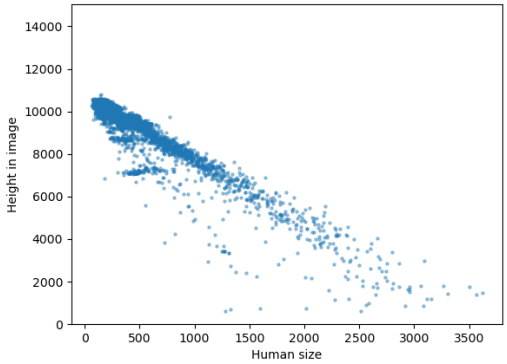

Fig. 7. Distribution of different-size objects on the vertical axis

Based on these investigations and experiments, I propose the main idea of the study. It is to take the size of the object in the image as the basis for judgment, and the image is partitioned and cropped. Then the image to be detected is processed by data compression. Before object detection, some methods are used to compress the amount of data in the video. The basis of partitioning is the range of human size detected in each region. Some regions have no human presence, and we can ignore that region. Some regions have small human size detected and we need to analyze them at their native resolution. Some regions have large human detections, and we can down-sample these regions to reduce the amount of computation. The downsampling multiplier depends on the value we set for the minimum accuracy. The compressed data is transmitted to the cloud for detection, and the inference results are obtained before fusing the results of each sub-image and outputting them to the original image.

- Select a stable and fast human detection model.
- Evaluate the relationship between human size and model performance to get a size vs. precision curve. Select a precision threshold (p) to get the minimum human size(S) to reach such precision.
- Split the image into patches and record the size range of all bbox containing the pixel point at each patch. (Calculated by extracting a few frames from the video).
- Obtain heat maps based on the human size range at each patch.
- Use minimum rectangle(s) to mark region(s) that contain all patches that human size $\in[0,2S)$ occur. We keep the original resolution to this type.
- Use minimum rectangle(s) to mark region(s) that contain all patches that human size $\in[2^nS, 2^{n+1}S)$ occur. We down-sample to $1/2^n * 1/2^n$.
- The compression-processed individual subgraphs are then transmitted to the cloud for inference and the inference results are sent to the edge devices.
- Finally, each inference result is fused, and post-processing such as NMS is performed.

3.2 Proposed Method

3.2.1 AP and Size Curve

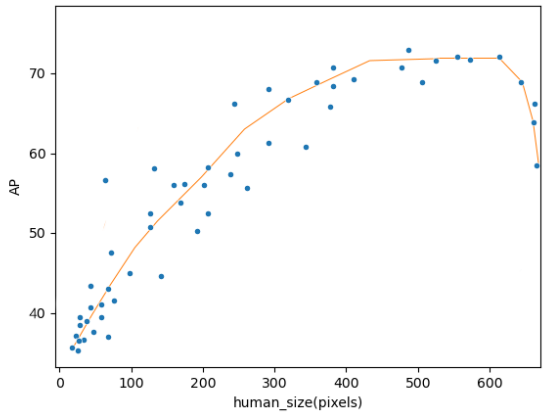

Fig. 8. Human size vs AP curve

After identifying the detection model yolov7 to be used in this framework, its performance is first evaluated. We deployed this model on different scenarios of the panda dataset and obtained detection results for objects of different scale sizes. The detection results are compared with the annotated boxes of the ground truth to obtain a set of data corresponding to the human size and accuracy. By fitting this set of data, the curve relationship between human size and AP value can be obtained, and with this set of data, we can judge the value of the smallest human size that is acceptable after downsampling processing after we set an accuracy threshold. Since the default input of yolov7 is 640*640 pixels, we limit the range of human size to 0-640 in the process of performance testing to facilitate the subsequent experiments.

3.2.2 Image Grids

Fig. 9. Grids in the example image

The purpose of the second step of this framework is to partition the image screen. Here we use one of the scenarios from the panda dataset as an example. First, we need to partition the original image into a grid, where the pixel value size of each grid is set to the input size after the default resize operation of the detection model. In this way, we can ensure the stability and reliability of the detection results.

3.2.3 Analyze the Composition of the Scene

Then we use the historical data to analyze the overall composition of the scene. We use the sliding window to crop the uncompressed original data and then input each sub-image into the detection model to be used to obtain the results. The inference results of each subgraph are then fused to obtain the detection data for the whole scene. This process is computationally intensive but belongs to pre-processing and can be done in the computationally rich cloud before the deployment of edge devices. The purpose of this step is to obtain detection frames with higher accuracy so that our subsequent analysis of the picture is more accurate, and the partition is clearer.

Fig. 10. Range of human size in each grid.

Then we count the range of human size that has appeared on each frame in the already partitioned grid and record it. Based on this range value, we can know which grids have larger pedestrian pixel frames, which grids have smaller pedestrian pixel frames, and which grids have no pedestrians appearing.

3.2.4 Image Partition

Using the statistical human size range of each grid, we can categorize the grids in the image by downsampling multiplicity. The first category is where no pedestrians have appeared, corresponding to the sky or buildings in the image; these grids do not need to be detected and therefore do not need to transmit their data to the cloud. (Fill color is gray in Fig9)

Each of the other colors corresponds to a different size range of grids, which are classified by different multiples of the minimum size value corresponding to the set accuracy threshold.

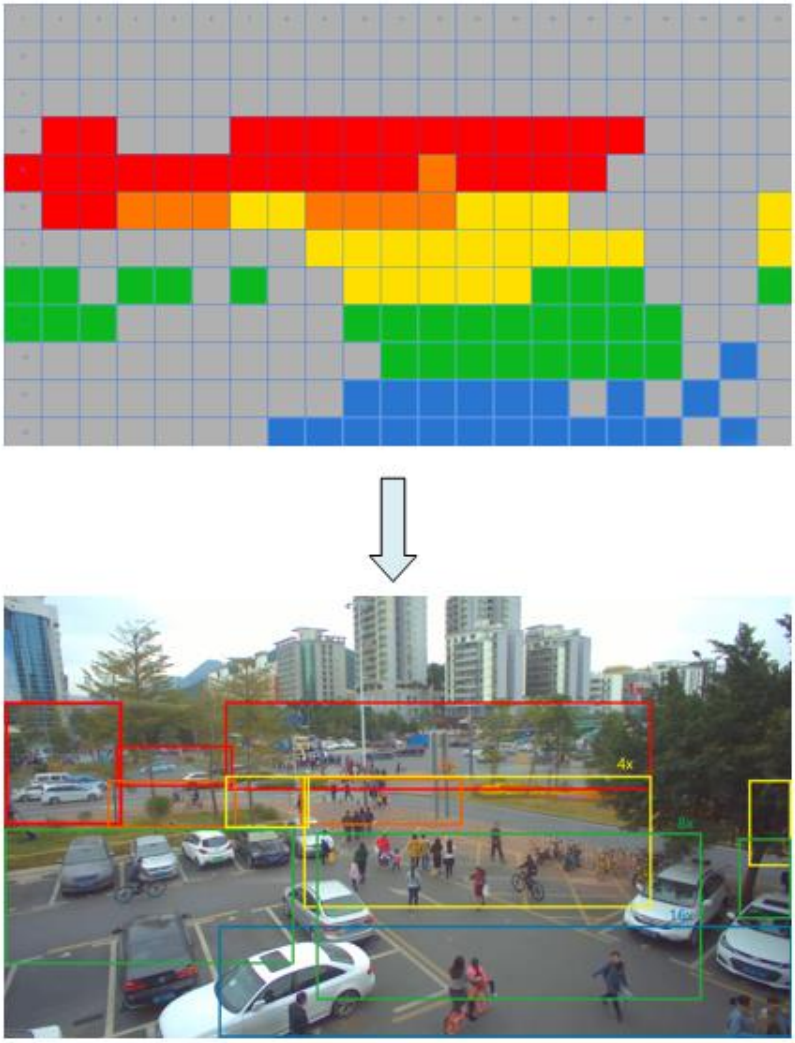

Fig. 11. Heat map and image partition

After obtaining the heatmap, the overall screen can be divided according to the coordinates of different grids. Some areas in this step are split into more reasonable cropping methods. Each region is expanded outward by 1/2*max(range) from the coordinates of its corresponding grid edge, which is the maximum human size value expanded outward by half. Since the size counting is done concerning the position of the center point of the detection box, this step of expanding the region is to ensure that the object to be detected appears intact in the divided region.

Region	Size level (the min-value in the range is greater than the min of size level)	Downsampling rate	Ratio to original data amount
Region0	/	/	0
Region1	0-2S	1	1
Region2	2S -4S	1/2 x1/2	1/4
Region3	4S -8S	1/4 x1/4	1/16
Region4	8S-16S	1/8 x1/8	1/64
Region5	16S-∞	1/16 x1/16	1/256

Table 1 Metrics on how to partition, the corresponding downsampling rate, and the radio to original data amount.

From Table 1, we can see the corresponding size level of different colors, the corresponding downsampling multiplicity, and the radio to the original data amount. From the data compression multiplicity, we can see that the larger the size level of the region, the smaller its down-sampled data after processing. Therefore, even if there is an overlap of different regions, the final data amount is smaller.

3.2.5 Data Compression

Fig. 12. Compressed image data amount vs. original image

In the process of data compression, we downsample the divided region. The downsampling approach used in the method proposed in this thesis is to incorporate a pooling layer that uses maximum pooling. This approach is to reduce the amount of computation while retaining more feature information.

In Fig10, the image in the foreground is the data that we need to transmit to the cloud for processing through the edge-cloud system connection. It can be seen that the amount of data to be transferred is greatly reduced.

During the transmission from the edge to the cloud, the amount of data that needs to be transmitted can be adjusted according to the actual arithmetic power of the edge, and since each image is compartmentalized, it can be very flexible to choose which data will be transmitted. The edge end can use a portion of the arithmetic that satisfies the processing conditions to perform detection on a portion of the image data. The advantage of this operation is that when there is sufficient computing power at the edge, the transmission step to the cloud can be eliminated as a way to again reduce the latency and increase the detection speed.

In cloud processing, each subgraph is detected using a sliding window. Note that the coverage area of each sliding window is set to have a 15% overlapping portion. In this way, objects at the edge parts of the image can be avoided to lose their feature information, resulting in omission in the detection results.

3.2.5 Post Processes

After the image input is detected in the cloud, its inference results, i.e., the detection frame data, are retransmitted to the edge device. The edge device then has to perform post-processing operations on the results again.

Firstly, the data results of the different sub-images are combined, and the detection frames located at the edges of the image are fused and adjusted.

Then NMS processing is also required at the edge end. In this step, we use soft

NMS as a classification method, which modifies the traditional NMS but does not increase the computational effort, which is the reason we chose it.

Firstly, candidate bounding boxes with confidence scores of the detected objects are generated and these candidates are sorted according to the confidence scores. Then, starting from the box with a high confidence score, each box is iterated, calculating the IoU values concerning the other boxes and decreasing their confidence scores accordingly. If the confidence score is below the threshold, this detection box is removed. After repeating the above operation for all candidate detection boxes, the remaining detection boxes obtained are the results of soft NMS.

The soft NMS allows potential detection frames to be retained while reducing overlapping results. The overall frame accuracy can be further improved in this way.

Chapter 4 Experiments

4.1 Experiments details

The graphics card used in the study is Nvidia RTX A6000 (Ampere GPU, 48GB GDDR6 memory, 768GB/s memory bandwidth). The code in this experiment was executed under the Ubuntu 20.04 operating system and written by Python 3.7.13, Pytorch 1.12.0.

To ensure the validity of the experimental results, we retrained the yolov7 model. The training dataset in the panda dataset was used during model training, considering the carrying capacity of the graphics card and the fairness of the results. We refer to the training scheme in baseline, which is to down-sample the original gigapixel images by a factor of 4 and use a 2,048*1,024 (pixels) sliding window to decompose the down-sampled images.

In the process of reasoning and testing the results, consider the baseline practice of downsampling the original images by a factor of 2. We also input the images into the detection framework proposed in this thesis after performing a 2-factor downsampling operation.

In the preliminary performance testing of yolov7, to test its effect under different conditions of scale, light, and occlusion range, we also used part of the dataset of pedestrians from the MOT challenge, but it was not used in the training and testing.

4.2 Experiment results

	AP	Average Time	FPS
Cascade R-CNN	0.30889	20s	0.05
Faster R-CNN	0.28442	14.29s	0.07
RetinaNet	0.22545	10s	0.1
YOLOv7(with slide window)	0.483	2.91s	0.337
Proposed Method	**0.412**	**0.486s**	**2.06**

Table 2 Results of our proposed method and the baseline

The Cascade R-CNN, Faster R-CNN, and RetinaNet results are provided as the baseline by the Panda dataset. In the table YOLOv7(with slide window) means that the input image was down-sampled by a factor of 2, and then a sliding window with a size of 640*640 pixels was used to do the detection.

The threshold requirement of AP is adjustable, when the lower the threshold requirement is set, the smaller the minimum value of human size in the compressed data, the higher the number of times the image can be compressed, and the faster the calculation speed? Therefore, in practical applications, the accuracy and inference time can be controlled and tuned.

In this experiment, the initial threshold of AP is set to 0.4, and according to the fitting curve of human size and AP we tested, we locate the minimum value of size at 64 pixels. and the scene image is divided into regions with S=64 pixels as the standard. In the experiment, we take one of the 30 frames of the video as historical data, which is used to analyze the scene. We used five scenes in the dataset to do the test and got the data results in Table 2.

It can be seen that our proposed scheme is higher compared to the preset AP threshold of 0.4. The reason for this is analyzed because there are some detection objects repeated within different subgraphs, so they are detected repeatedly, and their accuracy becomes higher after the post-processing soft NMS operation.

In addition to this, it can be seen that our proposed scheme outperforms the three models of baseline in terms of both accuracy and inference time. Although it cannot realize the real-time detection of the original yolov7 version, it also shows its efficiency with high-resolution images of gigapixels.

Chapter 5 Conclusion and future works

5.1 Conclusion

In summary, our research focuses on a tunable framework for pedestrian detection that is highly efficient in both accuracy and speed. It has several advantages as a proposed approach to solve the deployment problem on edge-cloud systems.

First of all, because our method divides the original data, it can support parallel computing of multiple small-scale pictures or videos.

Second, it is flexible to explore potential applications with other algorithms or detection models. The adaptability of this approach allows integration with existing detection models, enhancing its versatility and utility in real-world scenarios.

In addition, from the perspective of data transmission, the feasibility of computing some of the compressed data images directly on the camera side can be considered, while other images can be efficiently processed in the cloud. This dynamic division of computing between edge and cloud resources optimizes the balance of computing power and bandwidth, which can effectively alleviate resource constraints.

And, through our downsampling process, the target with a size change of more than a hundred times in the original image can be stabilized within a reasonable range, and this method also makes the detection result more stable.

Overall, our proposed method in this paper demonstrates its compatibility with various algorithms and its potential for super-resolution image object detection in edge-cloud systems.

5.2 Future works

The method proposed in this paper still has room for improvement, and these ideas may be implemented in future experiments. For example, although under the edge

camera, the composition of the image is stable, so that the trajectory of pedestrians can be traced. However, in actual deployment, the factors of the time axis can be considered. For example, at different time points, the areas where pedestrians appear may be different. Therefore, the area division method proposed in this paper can be optimized again.

In addition, due to the limited data set and validation set we used, some extreme cases may affect the validity of the test results, such as low light, or the situation where the target has a lot of occlusions. Therefore, this method can consider these factors, and there is still room for improvement.

As future research unfolds, this area may lead to the development of more robust and adaptable solutions for real-time object detection in high-resolution settings.

Reference

[1] X. Wang, X. Zhang, Y. Zhu, Y. Guo, X. Yuan, L. Xiang, Z. Wang, G. Ding, D. J Brady, Q. Dai, L. Fang, “PANDA: A Gigapixel-level Human-centric Video Dataset”, 10 Mar 2020, arXiv:2003.04852.

[2] C.Y. Wang, A. Bochkovskiy, H. Yuan and M. Liao: “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”, 6 Jul 2022, arXiv:2207.02696.

[3] Adam Van Etten, “You Only Look Twice: Rapid Multi-Scale Object Detection in Satellite Imagery”, 24 May 2018, arXiv:1805.09512.

[4] S. Jiang, Z. Lin, Y. Li, Y. Shu, Y. Liu: “Flexible high-resolution object detection on edge devices with tunable latency”, Pages 559–572, MobiCom 2021.

[5] N. Bodla, B. Singh, R. Chellappa, L. S. Davis,” Soft-NMS -- Improving Object Detection With One Line of Code”, 8 Aug 2017, arXiv:1704.04503.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin: “Attention Is All You Need”, 2017, arXiv:1706.03762.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby: “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, 2020, arXiv:2010.11929.

[8] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression”, 19 Nov 2019, arXiv:1911.08287.

[9] J. Xiao, T. Zhao, Y. Yao, Q. Yu, Y. Chen, “Context Augmentation and Feature Refinement Network for Tiny Object Detection”, 2021.

[10] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, P. Luo, “DetCo: Unsupervised Contrastive Learning for Object Detection”, arXiv:2102.04803.

[11] C. Yang, Z. Huang, N. Wang, “QueryDet: Cascaded Sparse Query for Accelerating High-Resolution Small Object Detection”, arXiv:2103.09136.

[12] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, P. Luo, “Sparse R-CNN: End-to-End Object Detection with Learnable Proposals”, 26 Apr 2021, arXiv:2011.12450.

[13] Z. Zong, G. Song, Y. Liu, “DETRs with Collaborative Hybrid Assignments Training”, 2 Jul 2023, arXiv:2211.12860.

[14] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, Y. Qiao, “InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions”, 17 Apr 2023, arXiv:2211.05778.

[15] J. O. Zhang, J. H. Cho, X. Zhou, P. Krähenbühl, “NMS Strikes Back”, 12 Dec 2022, arXiv:2212.06137.

[16] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, X. Chu,” YOLOv6 v3.0: A Full-Scale Reloading”, 13 Jan 2023, arXiv:2301.05586.

[17] A. Bochkovskiy, C. Wang, and H. Mark Liao, “YOLOv4: Optimal speed and accuracy of object detection.”, 2020, arXiv:2004.10934.

[18] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, 4 Jun 2015, arXiv:1506.01497.

[19] T. Lin, P. Goyal, R. Girshick, K. He, Pi. Dollár, “Focal Loss for Dense Object Detection”, 7 Aug 2017, arXiv:1708.02002.

[20] Z. Cai, N. Vasconcelos, “Cascade R-CNN: Delving into High-Quality Object Detection”, 3 Dec 2017, arXiv:1712.00726.

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer,” High-Resolution Image Synthesis with Latent Diffusion Models”, 20 Dec 2021, arXiv:2112.10752.

[22] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, J. Wang, “High-Resolution Representations for Labeling Pixels and Regions”,

arXiv:1904.04514

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. C. Berg,” SSD: Single Shot MultiBox Detector”, 8 Dec 2015, arXiv:1512.02325.

[24] K. He, G. Gkioxari, P. Dollár, R. Girshick, “Mask R-CNN”, 20 Mar 2017, arXiv:1703.06870.

[25] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, “Swin transformer v2: Scaling up capacity and resolution”. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[26] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. “Swin transformer: Hierarchical vision transformer using shifted windows.” In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)

[27] J. Fan, H. Liu, W. Yang, J. See, A. Zhang, W. Lin. "Speed up Object Detection on Gigapixel-level Images with Patch Arrangement", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022