UAV Autonomous Inspection System Based on Object Detection

August 2023

ZIRAN LI

Graduate School of

Science and Engineering

CHIBA UNIVERSITY

（千葉大学審査学位論文）

UAV Autonomous Inspection System Based on Object Detection

August 2023

ZIRAN LI

Graduate School of

Science and Engineering

CHIBA UNIVERSITY

Table of Contents

Acknowledgements

Time flies, and the years go by like a flash, and the short two years are just like a white horse passing by. At this point, it means that my doctoral career is coming to an end, and I look back on the two years I spent at Chiba University with gratitude.

First and foremost, I would like to express my sincere gratitude to my advisor, Prof. Akio Namiki, for his unwavering support, patience, and guidance throughout my doctoral studies. Prof. Namiki is serious and responsible, and every week we have a group meeting where students take turns to report on the progress of the week's work, and when we meet problems that are difficult to solve during the report, Prof. Namiki will patiently guide us and provide some constructive ideas and suggestions until the problems are solved, so that no student will be left behind.

Thanks to Prof. Wei Wang, who is very knowledgeable, at the beginning when I was confused about the subject, you guided me to the field of UAV and recommended me some excellent papers to study, which gave me the opportunity to get in touch with such a cutting-edge technology.

I would like to thank Prof. Satoshi Suzuki for giving me a lot of ideas to solve problems in my daily study, so that I can locate the problems quickly, and Prof. Suzuki's modest and rigorous working attitude has always influenced me.

I would like to thank Dr. Qi Wang, Dr. Hongxun Liu, Dr. Cheng Ju, Dr. Yanwen Zhang and Dr. Tianyi Zhang. During my two years of research career, you have helped me a lot and let me thrive. The time to discuss difficult problems with all of you is the most unforgettable.

Thanks to my family, you are my solid backing in the learning process, it is your support that allows me to face the difficulties bravely and let me find the right direction to strive for in life.

I would like to thank my alma mater, Chiba University, for giving me a full and wonderful 2-year doctoral career.

Finally, I send my most sincere thanks and respect to the scholars and experts who reviewed the papers.

Abstract

As the scale of the power grid continues to expand, the human-based inspection method is difficult to meet the needs of efficient power grid operation and maintenance, for this reason, research and development of fully autonomous overhead line inspection flight robot to achieve inspection of autonomous operations is of great significance.

(1) In order to realize the combination of object detection technology and UAV, a lightweight object detection model is designed based on YOLOv5. This model can be easily deployed on embedded devices, which lays the foundation for applying object detection technology to high-voltage power transmission line inspection.

(2) Aiming at the problems of low autonomy and low efficiency of intelligent identification of existing UAV transmission line inspection, an intelligent inspection system based on self-developed UAVs is designed, which improves the efficiency of transmission line inspection, while successfully integrating the object detection algorithm with the intelligent inspection system.

(3) In response to the problem that the target object deviates from the center of the picture when the UAV is flying autonomously at high altitude to take pictures of a specific object in (2), an autonomous UAV inspection system based on object detection is developed to correct the deviation, which greatly improves the quality of the dataset.

After a large number of flight verification, the intelligent inspection system greatly improves the efficiency of transmission line inspection, shortens the inspection cycle, reduces the investment cost of inspection manpower and material resources.

1 Introduction

1.1 Background

The power transmission line system is composed of a conductor system designed to transport electrical power from a power generating station to various distribution stations intended for residential and industrial purposes. While the underground transmission configuration is deemed more eco-friendly, its installation cost is notably higher than that of the overhead transmission system. As a result, overhead power lines are predominantly utilized for electric power transmission on a global scale[1,2]. These power lines which sometimes cut-across harsh environment (hot-desert, mountainous ranges, thick forest, and water bodies) are installed on vertically fixed towers using insulators, spacers, and dampers, among others[3]. Routine inspection of power transmission lines for early fault detection and maintenance is required for efficient and reliable transmission of high voltage power. The detection and location of transmission equipment faults is critical because it helps power transmission companies to minimize maintenance costs and prevent unnecessary outages[4]. As described in this paper[5], in the United States, a half-hour outage can cause an average loss of $15,707 to medium and large industrial customers, while an eight-hour outage can cause a loss of nearly $94,000. In addition, the growing global population and over-dependence on electricity supply necessitate the provision of effective strategies to examine power transmission lines. Figure 1-1 shows the annual spending costs of the major U.S. utilities on their distribution systems, and it can be seen that the annual spending costs are increasing each year.

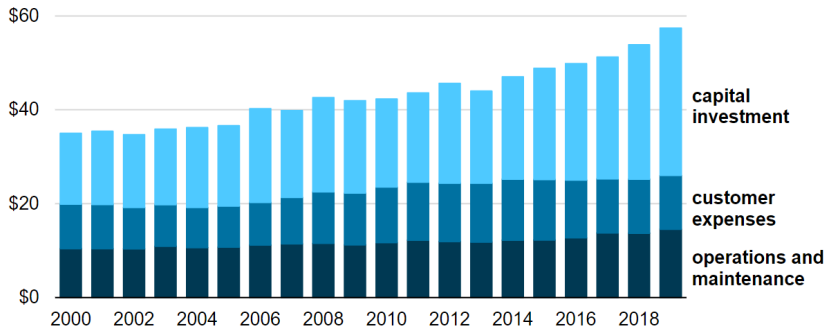


Figure 1-1: From 2000 to 2018, major U.S. utilities have been spending progressively more money on their distribution systems each year.[6]

Object detection stands out as one of the most fundamental and daunting problems in the field of computer vision, attracting extensive attention in recent years. In the past two decades, we have witnessed remarkable technological advancements in this field, which have profoundly impacted the entire domain of computer vision. If we regard current object detection technology as a deep learning-driven revolution, then we would observe the astute contemplation and long-term strategic design of early computer vision back in the 1990s[7]. Object detection represents a crucial computer vision task that pertains to the detection of visual object instances belonging to a specific category (e.g., individuals, animals, or vehicles) in digital images. The ultimate objective of object detection is to devise computational models and techniques that offer one of the most basic pieces of knowledge that computer vision applications require: what objects exist where? The two most vital metrics

for evaluating object detection algorithms are accuracy (inclusive of both classification accuracy and localization accuracy) and speed[7]. Target detection serves as the foundation for a multitude of other computer vision tasks, such as instance segmentation[8,9], image captioning[10], and object tracking[11]. In recent years, with the rapid growth of deep learning methods[12], the advancement of target detection has been considerably facilitated, leading to groundbreaking achievements and turning it into an unparalleled research hotspot. Currently, object detection is widely adopted in various real-world applications, including autonomous driving, robot vision, and video surveillance. Figure 1-2 manifests the increase in the number of publications related to "object detection" over the last two decades.

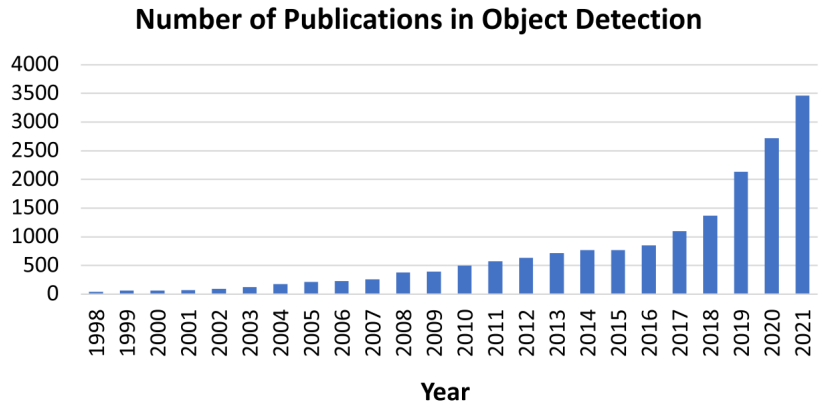


Figure 1-2: From 1998 to 2021, the number of publications in the area of object detection is increasing[7].

1.2 Goals of the Thesis

In order to meet the demand for intelligent and unmanned inspection robots for overhead line inspection, the fully autonomous overhead line inspection flying robot developed in this work mainly contains a quadrotor UAV and an artificial intelligence processing unit to perform operations; a cloud-based central station system for autonomous planning of cruise routes, naming of inspection data management, online fault diagnosis of returned data and remote operation; and an intelligent nest to automatically replace the operating inspection flying The intelligent nest, which is used to automatically replace and charge the robot's battery, is composed of three major parts. Mainly for the market existing inspection robot exist flight control stability, target object accurate collection, inspection robot intelligent recovery, inspection full autonomy, urban transmission line safety inspection and inspection data fault diagnosis and other problems to make certain technical breakthroughs.

1.3 Thesis Structure

By referring to domestic and international research on UAV transmission line inspection and object detection, analyzing and summarizing their advantages and shortcomings, and combining the development trend of inspection UAV transmission line inspection and object detection, the design framework of this thesis is proposed, as shown in Figure 1-3.

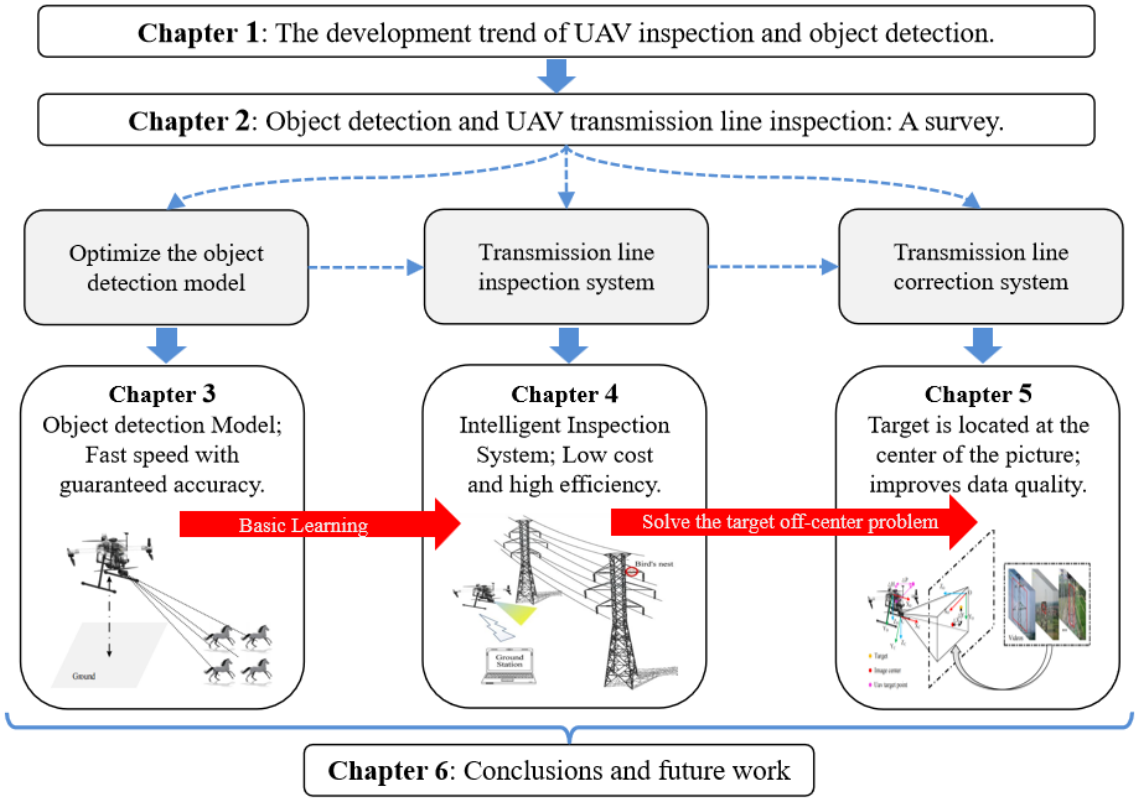


Figure 1-3: The structure of the thesis

Chapter 1 introduces the research background and development trend of UAV transmission line inspection and object detection. The importance of intelligent transmission line inspection is emphasized.

Chapter 2 introduces the development history of object detection and UAV transmission line inspection technology. In the new era of intelligent technology development, the combination of object detection and UAV transmission line detection is very necessary.

Chapter 3 designs a lightweight real-time object detector that can be deployed on an embedded platform for better integration with UAVs and is capable of real-time detection of grassland animals. This lays the foundation for the application of object detection technology to power transmission line inspection.

In order to solve the problem of high cost and low efficiency of transmission line inspection, Chapter 4 designs the automatic inspection system of UAV high voltage transmission line. The design idea and realization method of each function module in the system are also described.

Chapter 5 designs an autonomous UAV correction system based on object detection for the problem that the target object deviates from the center of the picture in Chapter 4. This greatly improves the quality of the data set and lays a firm foundation for later transmission line defect detection.

Finally, in Chapter 6, the design of this thesis is summarized, and the future design and research of this system are prospected.

2 Object Detection and UAV Power Transmission Line Inspection: A survey

2.1 Object Detection

In this section, we present a comprehensive overview of the history of object detection from various perspectives, including the evolution of milestone detectors, datasets, metrics, and key technologies.

2.1.1 The Evolution of Object Detection

During the last two decades, it has been widely acknowledged that advancements in object detection have been predominantly classified into two historical phases: the "traditional object detection phase (pre-2014)" and the "deep learning-based detection phase (post-2014)," as depicted in Figure 2-1. In the subsequent sections, we will provide a comprehensive summary of the pivotal detectors in this phase, utilizing the emergence time and performance as key indicators to emphasize the underlying technology driving them, as illustrated in Figure 2-2[7].

Milestone: Traditional detectors

Considering current object detection technology as a deep learning revolution, we can acknowledge the prescience and creativity of early computer vision in the 1990s. The majority of early object detection algorithms relied on handcrafted features. Since efficient image representations were lacking at that time, complex feature representations needed to be devised, along with diverse acceleration techniques.

Viola Jones Detectors

In 2001, Viola and Jones[13,14] introduced the first face detection algorithm capable of real-time detection without any constraints such as skin color segmentation. The Viola-Jones (VJ) detector used a sliding window approach that traversed all possible positions and scales in an image to detect faces. Although the detection process seemed simple, the computation required was beyond the capabilities of computers at that time. The VJ detector significantly improved detection speed by introducing three key techniques: the "integral image," "feature selection," and "detection cascade." By using these techniques, the VJ detector achieved comparable detection accuracy to other algorithms but was tens or even hundreds of times faster when running on a 700 MHz Pentium III CPU.

Histogram of Oriented Gradient (HOG) Detector

In 2005, Dalal and Triggs[15] introduced the Histogram of Oriented Gradients (HOG) feature descriptor, which can be seen as a significant improvement over the scale-invariant feature transformations[16] and shape context[17] methods of the time. The HOG descriptor strikes a balance between feature invariance (including translation, scaling, illumination, etc.) and nonlinearity by computing features on a dense grid of uniformly spaced cells and applying overlapping local contrast normalization

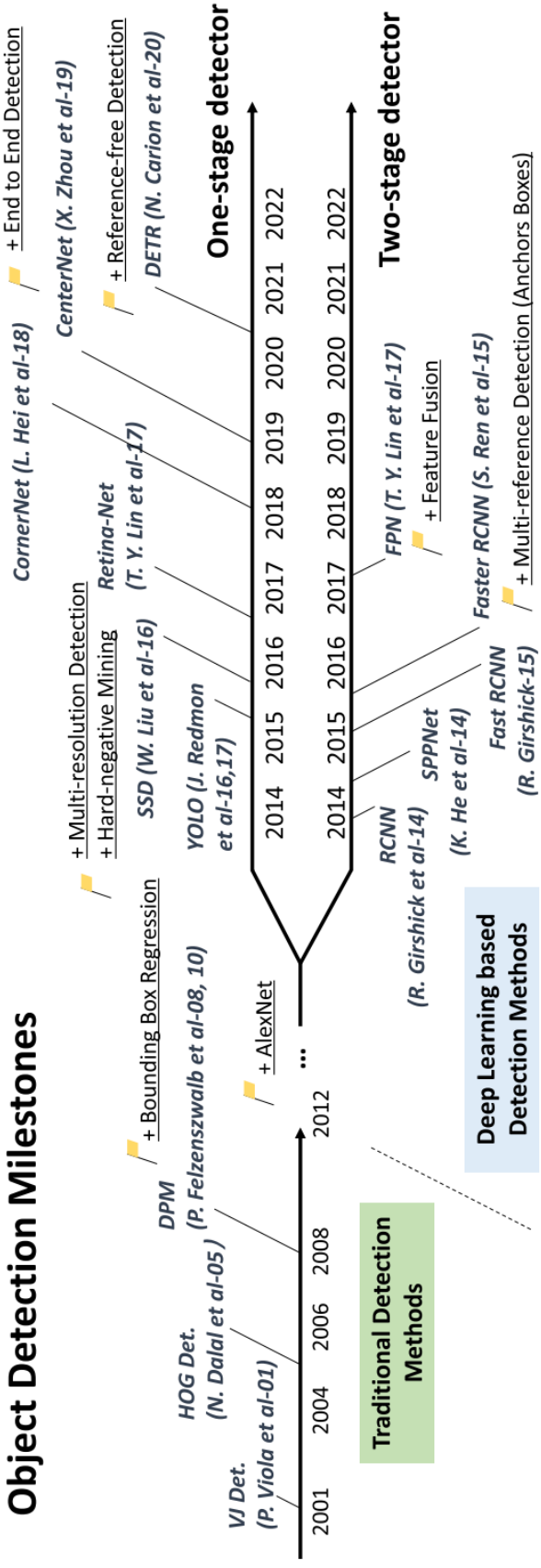


Figure 2-1: The Evolution of Object Detection.[7]

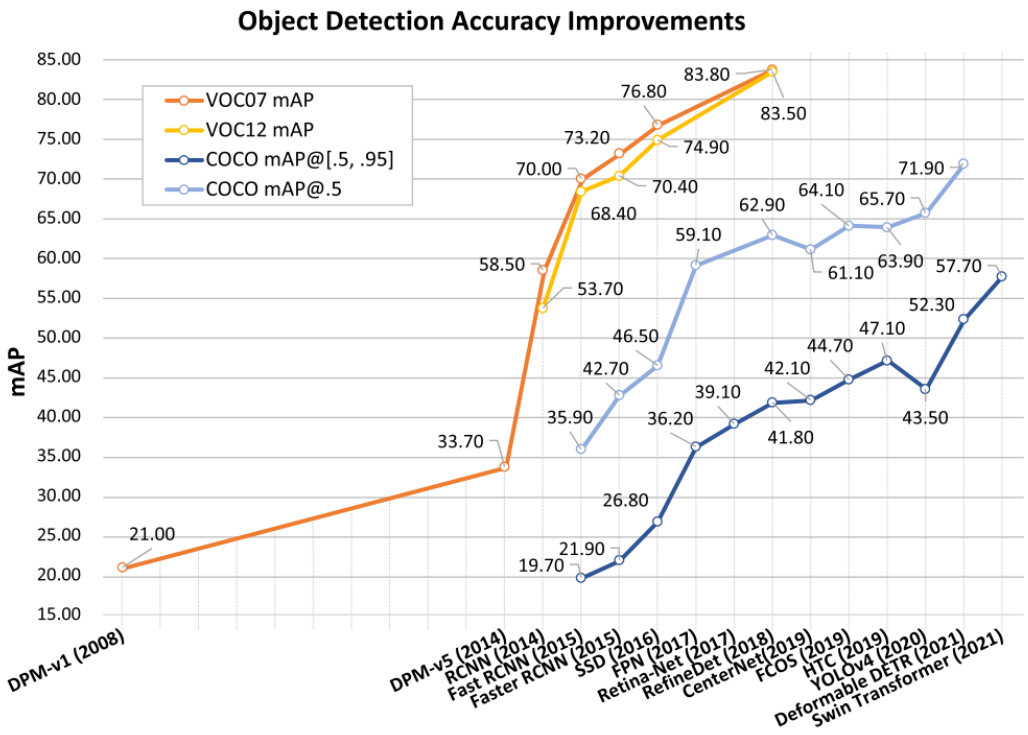


Figure 2-2: Progressive accuracy improvement of object detection on VOC dataset and COCO dataset.[7]

on "blocks". While HOG can be applied to detect various object classes, it was mainly developed for pedestrian detection. To detect objects of different sizes, the HOG detector rescales the input image several times while keeping the size of the detection window constant. HOG detectors have been a crucial component of many object detectors[18,19] and various computer vision applications for several years.

Deformable Part-Based Model

Deformable Part-Based Model (DPM) stands as a prime example of the classical object detection approach, having won the VOC-07, -08, and -09 detection challenges. It was initially introduced by Felzenszwalb et al[18] in 2008 as an extension of the HOG detector. Its detection principle follows a "divide and conquer" approach, where training involves learning how to appropriately decompose an object, while inference involves detecting different object parts. For instance, detecting a "car" can be decomposed into detecting its windows, body, and wheels, known as the "star model" proposed by Felzenszwalb et al.[18] Subsequently, Girshick[19,20,21] further extended this model to a "hybrid model" capable of handling objects with significant variations in the real world and made numerous other improvements. Despite today's object detectors surpassing DPM in terms of detection accuracy, they are still heavily influenced by its valuable insights, including mixture models, hard negative mining (HNM), bounding box regression, and context initiation.

Two-Stage Detectors

After a plateau in object detection research due to saturated performance of handcrafted features in 2010, convolutional neural networks (CNNs)[22] brought a resurgence in 2012. With the ability of deep CNNs to learn advanced and robust feature representations, the question arose: can we incorporate them into object detection? Girshick et al[23,24] pioneered this integration in 2014 with regions with CNN features (RCNNs), leading to unprecedented progress in object detection. In the deep learning era, there are two types of detectors: "two-stage detectors", which view detection as a "coarse-to-fine" process, and "one-stage detectors", which frame detection as a "one-step process".

Region-CNN

The concept behind Region-based Convolutional Neural Network (RCNN) is straightforward. It involves first extracting a set of region proposals (i.e., candidate bounding boxes) using selective search[25]. Each proposal is then resized to a fixed size and passed through a pre-trained CNN model on ImageNet, such as AlexNet[22], to extract features. Finally, a linear Support Vector Machine (SVM) classifier is employed to predict the presence of objects and classify them into different classes. RCNN achieved significant improvement on the VOC07 dataset, with mean average precision (mAP) increasing from 33.7% (using DPM-v5[26]) to 58.5%. However, it suffers from a major drawback of redundant feature computation for a large number of overlapping proposals, which results in slow detection speeds of up to 14 seconds per image when using a GPU. To address this issue,[27] was introduced in the same year.

Spatial Pyramid Pooling Network

In 2014, He et al.[27] introduced the Spatial Pyramid Pooling Network (SPPNet), which addressed the fixed-size input limitation of previous CNN models by incorporating a spatial pyramid pooling (SPP) layer[22] . This allowed the network to generate a fixed-length representation for regions of interest, regardless of their size or aspect ratio. Unlike R-CNN, which required redundant feature computation for each region proposal, SPPNet computed feature maps only once for the entire image, significantly reducing computation time. SPPNet achieved a VOC07 mAP of 59.2% while being over 20 times faster than R-CNN. However, SPPNet still had some limitations, such as its multi-stage training process and the fact that it only fine-tuned the fully connected layers. The following year, Fast R-CNN[28] was proposed, which addressed these limitations and achieved even higher detection accuracy.

Fast RCNN

Fast R-CNN was introduced by Ross Girshick[28] in 2015. This article is an improved version of R-CNN, with significant improvements in performance and computational efficiency. The main improvements over RCNN are in the following areas:

(1) Fast RCNN still uses selective search to pick 2000 suggestion frames, but here instead of inputting so many suggestion frames into the convolutional network, the original image is input into the convolutional network to get the feature map, and then the feature frames are extracted using the suggestion frames on the feature

map. The advantage of this is that the original suggestion frames overlap a lot and the convolution is repeatedly calculated, while here the convolution is calculated only once at each position, which greatly reduces the computation.

(2) Due to the different sizes of the proposed frames, the obtained feature frames need to be transformed into the same size, which is achieved by the ROI Pooling layer (ROI means region of interest i.e. target)

(3) There is no SVM classifier and regressor in Fast RCNN, the location and size of the classification and prediction boxes are output by convolutional neural network

(4) In order to improve the computational speed, the network finally uses SVD instead of fully connected layers

Faster RCNN

After the accumulation of R-CNN and Fast RCNN, Ross B. Girshick[29] proposed the new Faster RCNN in 2016, which structurally has integrated feature extraction, proposal extraction, bounding box regression (rect refine), and classification all integrated in one network, which makes the comprehensive performance improved, especially in the detection speed[30]. Faster RCNN can actually be divided into 4 main components:

(1) Convolution layer. As a CNN network target detection method, Faster RCNN first extracts feature maps of images using a set of basic conv+relu+pooling layers. the feature maps are shared for subsequent RPN layers and fully connected layers.

(2) Region Proposal Networks. The RPN network is used to generate regional proposals. this layer determines whether the anchors are positive or negative by softmax, and then uses the bounding box regression to correct the anchors to obtain the exact proposals.

(3) Roi Pooling. This layer collects the input feature maps and proposals and extracts the proposal feature maps after combining these information, which are sent to the subsequent fully connected layer to determine the target class.

(4) Classification. Using proposal feature maps to calculate the category of proposals, and again bounding box regression to obtain the exact final position of the detection box.

Despite the enhanced speed of faster RCNNs over fast RCNNs, computational redundancy still persists in the subsequent detection phase. In response, several advancements have been introduced, such as RFCN[31] and Light Head RCNN[32].

One-Stage Detectors

The majority of two-stage detectors follow a processing paradigm that progresses from coarse to fine. Coarse detection aims to enhance recall, while fine detection further refines localization by building on the results of coarse detection and prioritizing discriminative power. Although these detectors can achieve high accuracy without incorporating any elaborate features, their extensive complexity and slow processing speed make them an infrequent choice in engineering. In contrast, one-step detectors can identify all objects within a single inference step, making them popular for mobile devices with real-time and straightforward deployment features. However, their ability to detect small and densely-packed objects is notably compromised.

You Only Look Once

In 2015, Joseph et al[33] introduced You Only Look Once (YOLO), which is the first single-stage detector in the deep learning era[33]. It adopts a different approach from the two-stage detector and applies a single neural network to the entire image. The network segments the image into multiple regions and predicts the bounding box and probability of each region simultaneously. YOLO achieves remarkable detection speed, with the fast version running at 155 fps and achieving VOC07 mAP = 52.7%, while the enhanced version runs at 45 fps and achieves VOC07 mAP = 63.4%. However, YOLO has a localization accuracy disadvantage compared to the two-stage detector, especially for small objects. Later versions of YOLO[34,35,36] and the SSD, which was proposed later, are more concerned with this issue. Recently, the YOLOv4 team proposed YOLOv7[37], which surpasses most existing object detectors in terms of speed and accuracy, ranging from 5 to 160 fps. It achieves this by introducing optimized structures such as dynamic label assignment and model structure reparameterization.

Single-Shot Multibox Detector

In response to the respective shortcomings and advantages of YOLO and Faster R-CNN, WeiLiu et al.[38] proposed Single Shot MultiBox Detector, referred to as SSD. The whole network of SSD adopts the idea of one stage to improve the detection speed. The network also incorporates the idea of anchors in Faster R-CNN, and does feature extraction in layers and computes border regression and classification operations sequentially, which makes it possible to adapt to the training and detection tasks of multiple scales of targets. The idea of SSD network subject design is feature extraction in layers, and edge regression and classification in turn. Because different levels of feature maps can represent different levels of semantic information, low-level feature maps can represent low-level semantic information (containing more details), which can improve the quality of semantic segmentation and are suitable for learning small-scale targets. High-level feature maps can represent high-level semantic information, smooth segmentation results, and are suitable for in-depth learning of large-scale targets. Therefore, the network of SSD proposed by the authors can be theoretically suitable for target detection at different scales.

CenterNet

Zhou et al[39] introduced CenterNet in 2019, which adopts the keypoint-based detection model, but dispenses with costly post-processing steps such as group-based keypoint assignment (as seen in CornerNet[40] and ExtremeNet[41]) and non-maximum suppression (NMS), resulting in an end-to-end detection network. In CenterNet, objects are represented as a single point, the center of the object, and all its properties, including size, orientation, position, and pose, are regressed based on a reference centroid. This approach is both simple and elegant, and enables the integration of multiple tasks, such as 3D object detection, human pose estimation, optical flow learning, and depth estimation, into a single framework. Despite its concise detection concept, CenterNet achieves comparable detection results, with COCO mAP0.5 at 61.1%.

Detection Transformer

Before Detection Transformer (DTER)[42], target detection in the field of deep learning can be roughly divided into: one-stage detection and two-stage detection. However, none of these detection methods can directly obtain the detection results (first, a dense proposal is used to cover the part of the whole image where the object may appear, and then the category information is predicted, adjusted, and the position information is obtained; in simple terms, it is similar to a jigsaw puzzle, in which a small piece of the whole puzzle is detected first, and then each small piece is put together, and then these small pieces are adjusted until the image from these small pieces The image is similar enough to the given ground truth, i.e., it satisfies some artificially set threshold); and the previous detection method has some problems, such as: repeating a large number of prediction frames, the design of anchors, how to heuristically assign the target to be targeted to several anchors, etc.; to solve these problems usually requires some processing, such as NMS, anchor generation, etc., but these operations can greatly affect the performance of detection. Compared with the previous target detection, DETR is a more intuitive approach. DETR finds targets similar to finding targets in a map, first roughly searching the global scope, and then precisely locking the target with a magnifying glass; therefore, its detection of small objects is not very effective. Macroscopically speaking, the previous detection is a way to go from small to large, from local to global, gradually integrated, and then find the target; while DETR is a way to go from large to small, from global to local[43].

2.1.2 Technology development in object detection

In this section, we introduce several crucial components of detection systems and their technological advancements. These components encompass multiscale detection, Context Priming, and Nonmaximum Suppression.

Technical Evolution of Multiscale Detection

Multiscale detection is a critical technical challenge in detecting targets of varying sizes and aspect ratios. Over the last two decades, multiscale detection has undergone several historical stages, as depicted in Figure 2-3.

Feature Pyramid and Sliding Window

Following the VJ detector, researchers shifted their focus to a more intuitive approach to detection, which involved constructing "feature pyramids and sliding windows." These windows were typically of a fixed size, and less consideration was given to "different aspect ratios." In order to detect objects with a more complex appearance, Girshick et al. explored better alternatives beyond the feature pyramid. The "hybrid model" [20] was one such solution at that time, where multiple detectors were trained for objects with different aspect ratios. Additionally, sample-based detection[44,45] offered another alternative by training a separate model for each object instance.

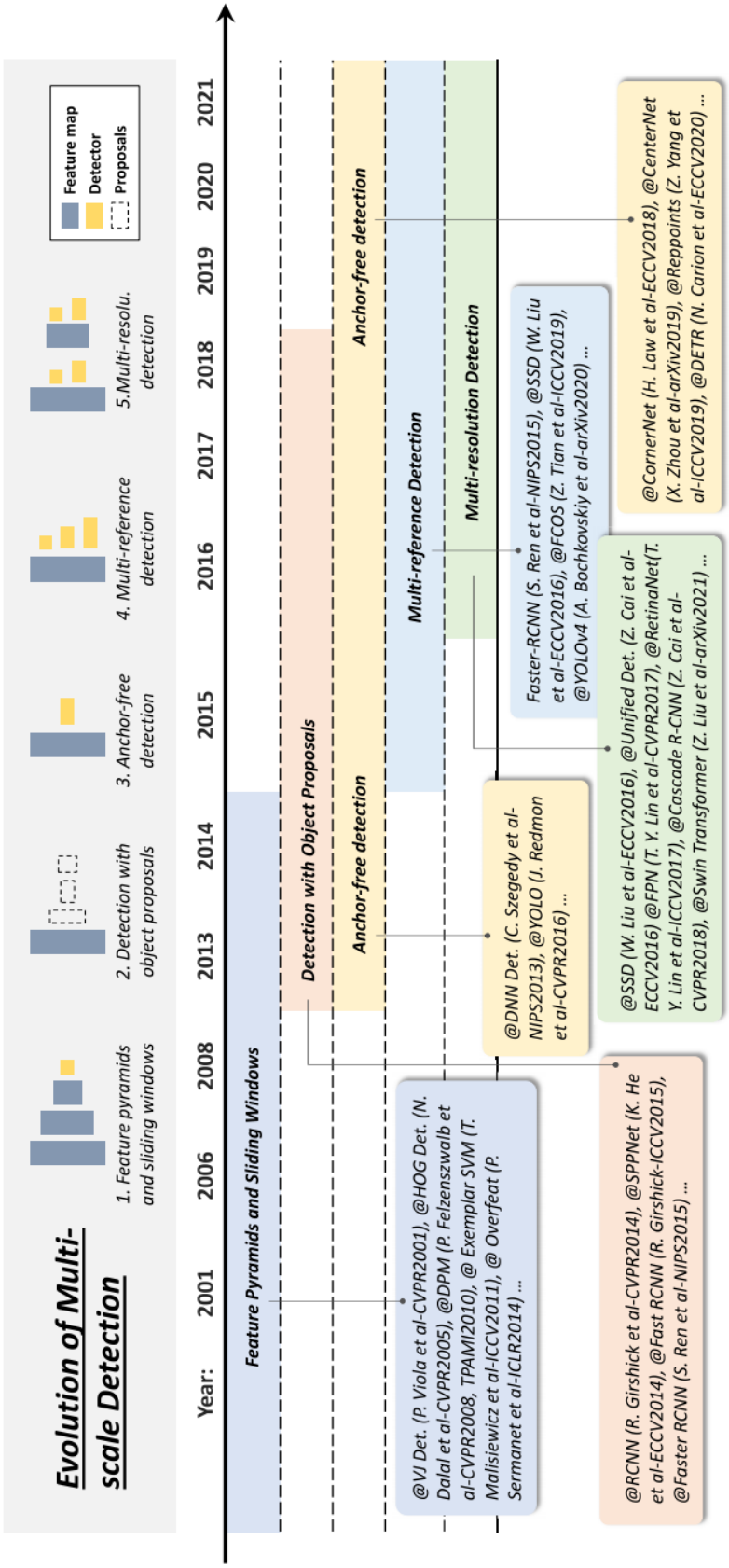


Figure 2-3: The development of multiscale detection techniques in object detection. [7]

Object proposals

Object proposals are a collection of class-agnostic frames that have the potential to contain any object. Their use in object detection allows us to avoid an exhaustive sliding window search across an image. For a thorough overview of this topic[46,47], we recommend the reader consult the following publications. Initially, proposal detection methods adopted a bottom-up detection philosophy[48,49]. However, since the advent of deep CNNs in visual recognition, top-down, learning-based approaches have demonstrated a greater edge in this area since 2014[50,51]. Currently, with the advent of one-stage detectors, proposal detection is slowly losing prominence.

Anchor-Free Detection

In recent times, the task of multiscale detection has become less complex and more direct with the increasing availability of GPU computing power. The concept of solving multiscale issues through deep regression has become simpler, wherein the bounding box coordinates are predicted based on the deep learning features directly[52]. Subsequently, since 2018, researchers have been contemplating object detection in terms of keypoint detection. These methods are typically based on two ideas: one is a group-based approach that identifies keypoints such as corner points, centroids, or representative points and then performs per-object grouping[40,41,53,54]; the other is a group-free approach that considers an object as one or multiple points and then regresses object properties such as size, scale, etc., with reference to these points[39,55].

Multireference/Multiresolution Detection

The approach of multi-reference detection has gained significant popularity and is widely used in the field[29,35]. The underlying concept of this approach is to establish a set of reference points, known as anchor points, which includes both boxes and points, at every position in the image. The goal is to predict boxes based on these reference points. On the other hand, multi-resolution detection[38,56,57] is also a commonly employed technique that involves detecting objects at varying scales across different layers of the network. Both multi-reference and multi-resolution detection have emerged as crucial components of modern target detection systems.

Technical Development of Context Priming

Visual objects are commonly found within specific contextual environments. The human brain utilizes these contextual connections to facilitate visual perception and cognition[58]. Background priming has been a long-standing technique used to enhance object detection. The evolution of background priming in object detection is illustrated in Figure 2-4.

Detection With Local Context

Local context, referring to the visual information in the vicinity of the object of interest, has been widely recognized as a crucial factor in improving object detection.

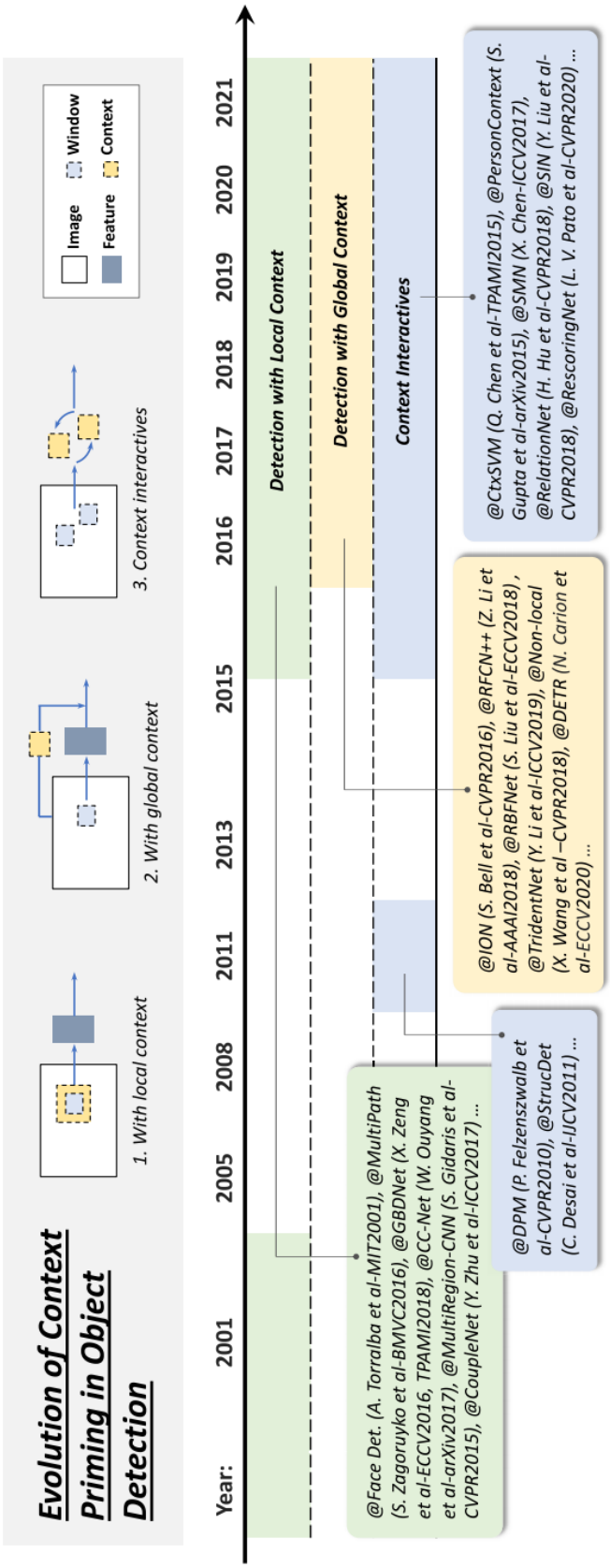


Figure 2-4: Evolution of context priming in object detection.[7]

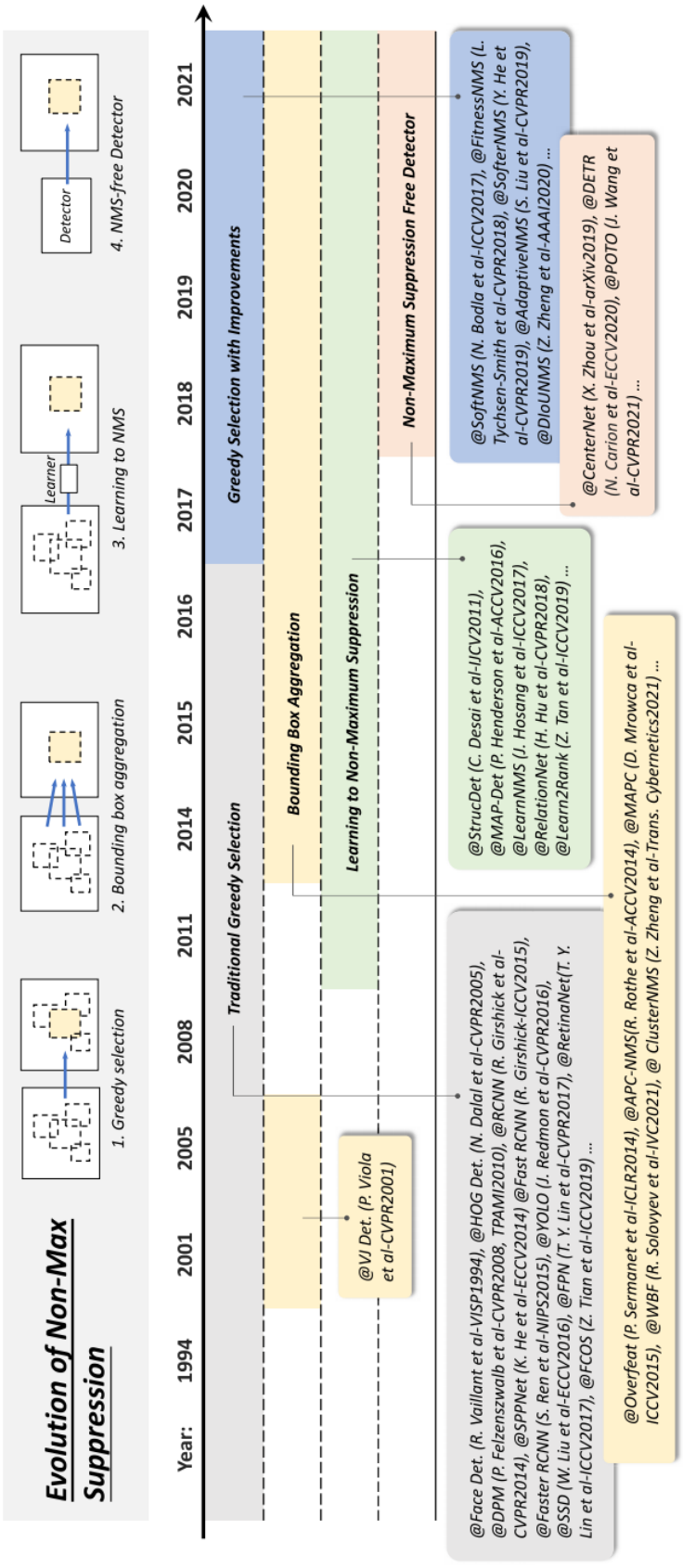


Figure 2-5: History of non-maximum suppression (NMS) techniques in object detection.[7]

In the early 2000s, Sinha and Torralba[59] showed that incorporating local contextual regions such as facial boundary contours could substantially enhance facial detection performance. Additionally, Dalal and Triggs[14] found that even a small amount of background information could lead to more accurate pedestrian detection. Recent advances in deep learning-based detectors have demonstrated the effectiveness of exploiting local context through methods such as expanding the perceptual domain of the network or enlarging the object proposals[60,61,62,63,64,58].

Detection With Global Context

Global context refers to using information about the overall scene to aid in object detection. Early detectors used statistical summaries, such as Gist[58], to integrate global context. In recent years, there are two main approaches for integrating global context. The first approach uses operations like deep convolution, extended convolution, deformable convolution, and pooling to obtain a larger receptive field[65,66]. Attention-based mechanisms, such as non-local mechanisms and transformers, have also been successful in capturing global context information[67]. The second approach considers global context as sequential information and uses recurrent neural networks to learn it. By leveraging global context, object detectors can improve their performance and better understand the relationships between objects and their surroundings[68,69].

Contextual Interaction

Contextual interactions refer to the constraints and dependencies shared among visual elements. Recent studies have demonstrated that incorporating contextual interactions can enhance the performance of modern object detectors. Two categories of approaches have been proposed to leverage contextual interactions. The first category explores the relationships between individual objects[70,71,72,73], while the second focuses on dependencies between objects and the surrounding scenes[74,75].

non-maximal suppression

As nearby windows frequently yield comparable detection scores, non-maximal suppression (NMS) is utilized as a post-processing measure to eliminate redundant bounding boxes and obtain the ultimate detection results. However, in the early days of target detection, NMS was not always incorporated due to the incomplete understanding of the desired output of a target detection system[76]. Figure 2-5 visually depicts the evolution of NMS over the past two decades.

Greedy Selection

The traditional approach for performing NMS, although widely used, is considered outdated. This approach involves selecting the bounding box with the highest detection score from a set of overlapping detections and removing its neighboring bounding boxes based on a pre-defined overlap threshold. However, this approach has limitations, such as potentially selecting an unsuitable box with the highest score, suppressing nearby objects, and failing to suppress false positives[77]. As a

result, there have been various proposed improvements to overcome these limitations[78,79,80].

Bounding Box Aggregation

Bounding Box (BB) aggregation is a set of techniques used for object detection that aims to combine or cluster multiple overlapping bounding boxes into a single final detection. This approach offers an advantage over traditional non-maximum suppression (NMS)[81] techniques as it takes into consideration the spatial layout of objects and their relationships[82,83]. Several successful detectors, such as the VJ detector and the Overfeat, which was the winner of the ILSVRC-13 localization task, have incorporated this method into their framework[84].

Learning-Based NMS

Learning-based NMS has emerged as a promising approach to improve the performance of object detection systems, particularly in scenarios involving occlusion and dense target detection[85,86,87,78]. Unlike traditional manual NMS methods, learning-based NMS treats NMS as a filter to rescore all original detections and trains the NMS as part of a network in an end-to-end manner or trains a network to mimic the behavior of the NMS. This approach has garnered significant attention in recent years and has demonstrated impressive results.

NMS-Free Detector

Researchers have recently developed a series of methods aimed at achieving one-to-one label assignment, which means assigning only one prediction box to each object and avoiding the need for NMS altogether[42,88]. These methods often involve training with high-quality boxes and following specific rules to achieve NMS-free detection. NMS-free detectors, which are more similar to the human visual perception system, represent a promising direction for the future of object detection.

2.2 UAV Transmission Line Inspection

2.2.1 The Evolution of UAV Transmission Line Inspection

The advancement of new-generation information technology has led to artificial intelligence, cloud computing, and big data emerging as potent drivers of smart grid technology. In the quest to minimize human involvement in transmission line inspection, researchers have turned their attention towards devising autonomous inspection methods with high intelligence, presence, and reliability. In the ensuing sections, we elucidate the evolution of power inspection and the UAV inspection system[89].

Traditional Transmission Line Inspection

To guarantee the secure and dependable operation of transmission lines and their accompanying equipment, regular inspection of transmission lines is a fundamental responsibility of the grid system[90]. This includes high-voltage towers, auxiliary

components, transmission lines, and transmission channels, among other things. In traditional manual inspections, personnel must climb the towers for inspection As shown in Figure 2-6.. This is a labor-intensive and time-consuming process, and the energized state of the tower and the complex environment can pose significant safety risks. Additionally, some sections of the transmission line are restricted by terrain factors, which makes inspection difficult or impossible[89].

To overcome these challenges, grid operators have turned to artificial intelligence technologies in recent years. UAV inspection[91,92,93], robotic inspection[94,95], and helicopter inspection[96,97] are all widely used to inspect overhead transmission lines using sensors[98] such as cameras and infrared images. Robotic inspection, however, has limited coverage and hidden operation, and routine maintenance is difficult. Helicopter inspection is expensive and has strict site requirements. While UAV inspection's accuracy is not as high as other technologies, it offers high portability, mobility, safety, and efficiency. Consequently, it is frequently used in combination with manual inspection to enhance the accuracy and safety of the power system[89].

Figure 2-6: Manual inspection along transmission lines.

UAV Transmission Line Inspection

In recent times, the detection of transmission lines using unmanned aerial vehicles (UAVs) has garnered significant attention from researchers[99]. As depicted in Figure 2-7, various autonomous flight control algorithms have been proposed to enable precise flight control of UAVs[100,101]. Additionally, extensive research has been conducted on UAV guidance systems[102] and nonlinear control systems[103,104]. Notably, in 2015, the University of Technology in Madrid developed a new type of UAV that utilizes a visual navigation system for autonomous flight inspection missions, with GPS position correction enabled[105]. Similarly, in 2016, a Japanese R & D team developed a specialized UAV combat system that accurately locates and identifies insulator finches, bird nests, and fallen branches[106].

Furthermore, several research proposals have been put forward for UAV-based transmission line inspection, including those by Pouliot et al.[107] and Zhang et al.[108], which propose image-based automatic detection methods that can detect various

aspects of transmission lines. However, detailed inspections of critical components, such as insulators and accessories, are relatively rare. Tang et al.[109] utilized GPS and image data processing algorithms, along with tracking techniques, to detect information regarding the UAV's location relative to a reference object, thereby allowing a general condition check of the transmission line. However, detailed inspections of critical components, such as insulators and accessories, cannot be performed using this method.

To overcome these limitations, researchers have employed improved radon transform and curvelet transform (CRT) methods for extracting linear features from satellite images. CRT is especially robust against random noise and system noise caused by nonlinear features[89]. In 2017, Manohar[110] employed a mobile LiDAR device to extract power lines. The method successfully extracted partially obscured power lines with an average accuracy and completeness of 98.84% and 90.84%, respectively.

More recently, Nguyen et al.[111] summarized the latest methods and theories for automatic vision-based power line detection systems and discussed the challenges and possible solutions. However, with the increasing popularity of drone inspection technology, new challenges have emerged. There is a need for highly skilled UAV inspectors, as their technical abilities directly impact the quality of inspections and the safety of electrical facilities. The lack of professional UAV patrol personnel and the limited intelligence of UAV inspection systems are the primary limiting factors for the promotion and application of UAV inspections[89].

Figure 2-7: UAV transmission line inspection.

Transmission Line Inspection Based on Image Processing

During the inspection process, a large volume of image data is generated, particularly for visible light data, which necessitates manual processing and relies on human inspection to identify and mark defective images. This traditional approach is not only inefficient, but also poses the risk of increasing the false detection rate. To address this, numerous researchers have attempted to employ image processing techniques to automatically analyze inspection images and identify potential defects, as

illustrated in Figure 2-8. Artificial intelligence has emerged as a prominent research area in recent years, and has demonstrated remarkable achievements in diverse fields including image detection[112], speech recognition[113], and data analysis[114]. In particular, deep learning techniques have become the dominant approach in the image processing field[115]. These methods eliminate the need for manual feature extraction and are better suited to handling visible images obtained during power patrol, which often feature complex backgrounds, variable scenes, and diverse target features. To identify surface breakages in insulators, a multi-scale residual neural network was proposed[116]. This approach employs three convolutional kernels of varying sizes for convolutional filtering and feature image fusion, enhancing the spatial and channel correlations of the feature maps. Additionally, a seven-layer convolutional neural network (CNN) was used to accurately detect the partial discharge status of high-voltage cables with an accuracy rate of 92.57%, surpassing that of support vector machines and back propagation neural networks[117].

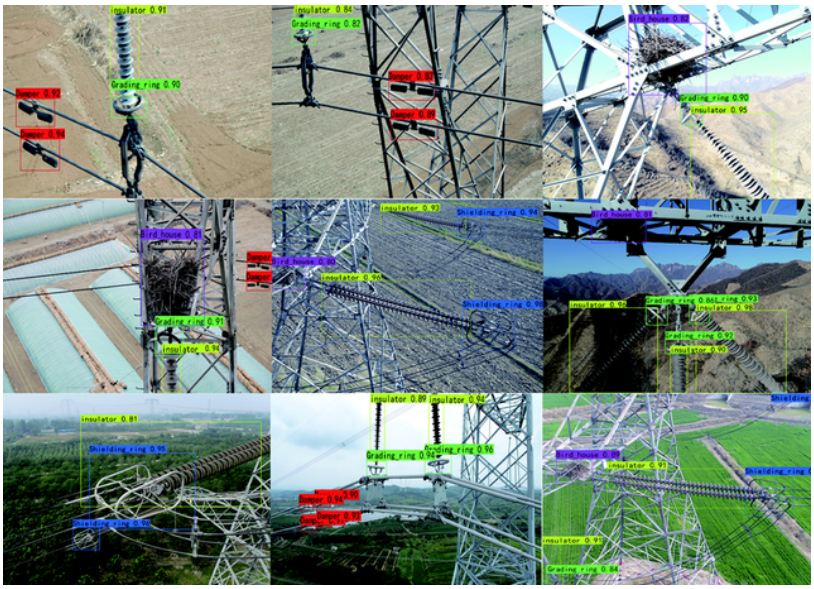


Figure 2-8: Transmission line detection based on image processing.

2.2.2 The System of UAV Transmission Line Inspection

The UAV-based electric power inspection system comprises six main components: the UAV itself, a navigation module, a trajectory tracking module, a Pan/Tilt/Zoom (PTZ) detection module, a ground station, and a power supply module. The UAV subsystem is equipped with a PTZ and other equipment, and its range and stability are critical factors determining the feasibility of the inspection system. Navigation technology is pivotal to enable autonomous inspection by the UAV, and is comprised of DGPS and inertial navigation. Given the complexity of power line distribution environments and the need for precise UAV photo positioning, navigation accuracy and reliability are essential to achieving autonomous inspection. The tracking technology plays a vital role in ensuring the UAV reaches designated target points for effective detection. Deviation of the UAV from the target point may impede detection system's ability to locate the target. The PTZ detection subsystem is a critical component for enabling autonomous detection, consisting primarily of PTZ

and image recognition modules. The ground station subsystem primarily monitors the UAV status and transmits real-time image information, while also being capable of sending control commands to the UAV. The power supply subsystem provides the necessary energy for the UAV to conduct inspections in an orderly fashion. Figure 2-9 in[89] illustrates the structural composition of the UAV inspection system.

UAV

The UAV subsystem is a critical component of the UAV electric power inspection system, comprising of the airframe and the UAV flight control module. The airframe serves as a platform to carry equipment and payload, while the flight control module is the core of the UAV system, responsible for controlling the flight status of the UAV. The flight control module performs a range of crucial functions, such as adjusting the flight attitude in response to user commands, regulating the attitude and capturing images with the PTZ equipment. Additionally, the flight control module offers secondary development capabilities. It can obtain control authority of the UAV through the communication port linked to the combat control board and control the status of the UAV and PTZ. The flight control module is capable of real-time communication with the ground station, transmitting the collected data and image information while accepting and executing commands.

Navigation Module

To enhance the accuracy and stability of navigation, combined navigation module[118,119,120] is employed. This technique utilizes multiple positioning systems to measure the same information source, and extracts and corrects errors of each system from the comparative values of these measurements. Kalman filtering is used to fuse differential GPS data and INS position information, resulting in accurate position information. The combined differential GPS/INS navigation principle is illustrated in Figure 2-10.

Trajectory Tracking Module

Correct path tracking is essential for the successful completion of a UAV inspection mission. The flight control module plays a crucial role in achieving this objective. Once the UAV path planning is completed and uploaded to the flight control module, the navigation module guides the UAV to accurately reach the planned cruise point on the desired route. Real-time position information of the UAV is continuously transmitted to the flight control module. The flight control module compares this information with the planned cruise point, and if there is any discrepancy, it sends a control command to the UAV to correct its position. This iterative process continues until the UAV reaches the cruise point and completes its mission. Researchers have proposed various path planning and tracking algorithms to improve the performance of UAV inspection systems[121,122]. These algorithms use advanced machine learning techniques and optimization methods to generate optimal paths for the UAV and ensure accurate path tracking.

PTZ Detection Module

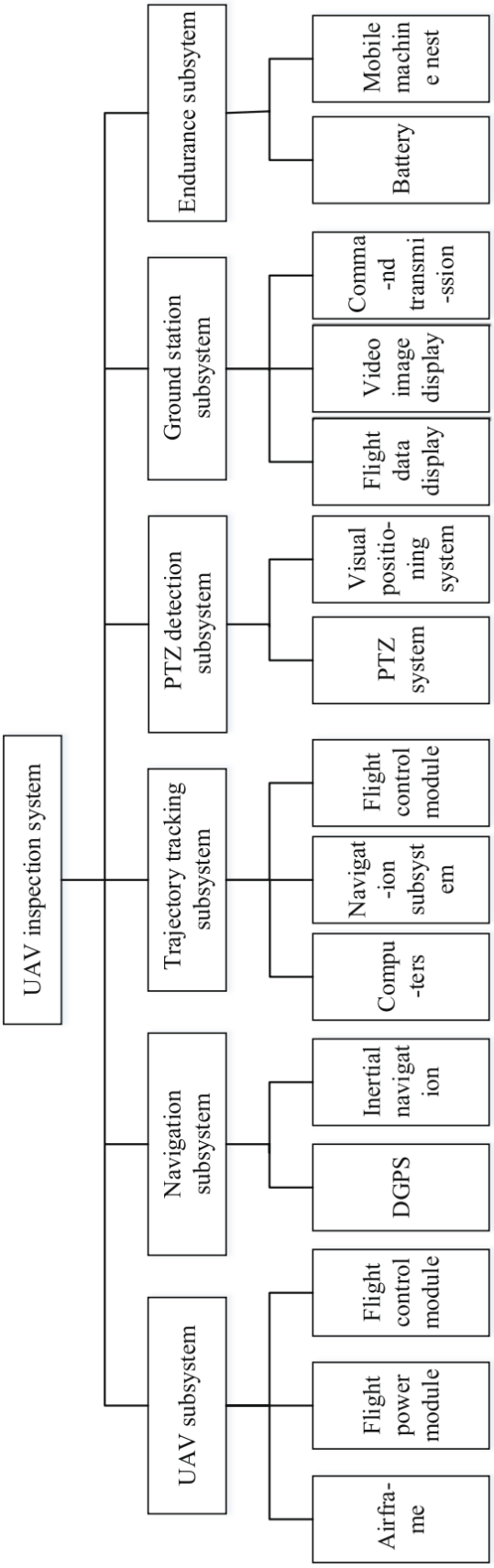


Figure 2-9: Composition structure of the UAV transmission line inspection system.

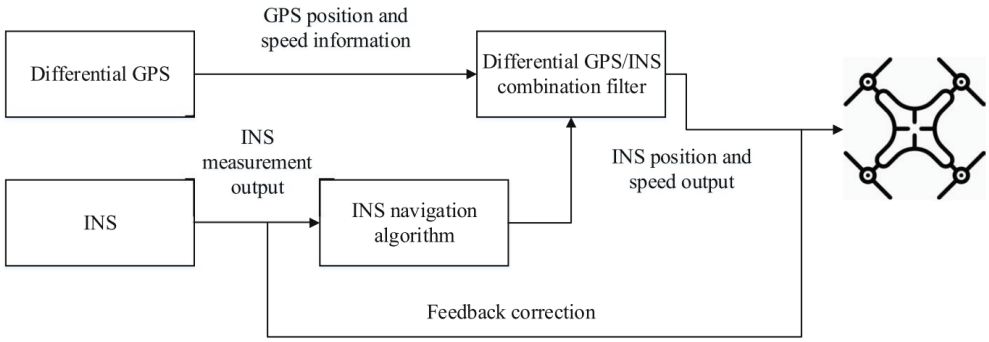


Figure 2-10: Integrated navigation principle of diferential GPS/INS.

In order to more accurately detect any underlying issues in transmission equipment, the PTZ detection module is implemented[123,93]. This module is comprised of PTZ and image recognition modules. The PTZ component[124,125] receives commands from the flight control module, which includes adjustments to the camera's orientation and capturing images. The image recognition module includes a vision sensor, a dedicated image processor, and a computer. Its main function is to identify targets in the acquired images. Based on the recognition results, the computer sends subsequent commands to the PTZ and the UAV through the flight control module. Figure 2-11 depicts the configuration of the image recognition module.

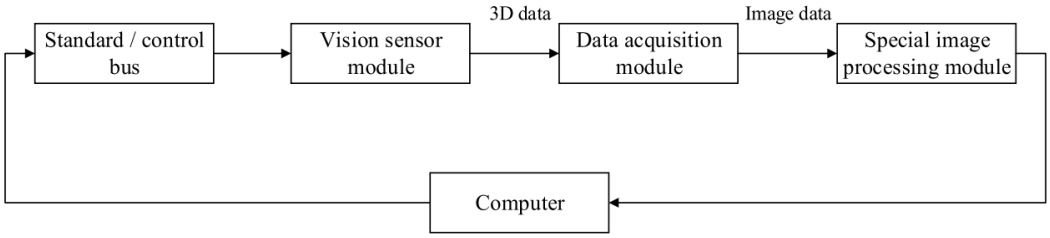


Figure 2-11: Structure of the image recognition module.

Ground Station

The ground station plays a pivotal role in the inspection process as it facilitates monitoring and adjustment of the multi-rotor UAV's flight dynamics. Functioning as a graphical user interface, it furnishes the user with a platform to manage combat missions and parameters. By using the ground station, the user can manually oversee the UAV, relay pertinent information, and even directly manipulate flight parameters[126,127]. Typically, in practical applications, a radio modem is employed to connect the UAV to the ground station.

Endurance Module

One of the major limitations hindering the effectiveness of UAV inspections is insufficient endurance, which also constitutes a crucial challenge that must be tackled to achieve fully autonomous UAV inspections. The current power inspection by UAVs relies predominantly on small multi-rotor aircraft with a typical range of

20-45m, and even for medium and large aircraft that employ an oil-electric mix, it is challenging to exceed 3 hours of flight time. The need for frequent battery replacement seriously impedes the efficiency of power inspection. To address this problem, besides augmenting the battery capacity and minimizing the power consumption of the UAV, deploying the UAV nest has become a popular solution. Coupled with the UAV's autonomous takeoff and landing technology, this approach enhances the automation of UAV battery management, thereby mitigating the impact of range limitations on patrol efficiency[89].

2.3 Conclusion

This chapter introduces the main techniques in object detection and the components of the UAV transmission line inspection system. In chapter 3, 4 and 5, the object detection technology is involved, but the necessary technology is not explained in detail. Similarly, the UAV transmission line inspection system is designed in chapter 4, 5 and **??**, although relevant explanations are also carried out, but not so specific. Therefore, we introduce the object detection technology and each component of transmission line inspection system in detail in this chapter. This enables the reader to have a clear understanding of object detection techniques and UAV inspection systems, and lays a knowledge foundation for the reader to read the subsequent chapters.

3 Application of Low-Altitude UAV Remote Sensing Image Object Detection Based on Improved YOLOv5

3.1 Introduction

In recent years, in improving the protection of grassland wildlife, it is essential to determine the number and distribution of grassland animals[128]. . Traditional manual methods of obtaining statistics are slow and dangerous. Therefore, in the field of artificial intelligence, especially in the continuous development of computer vision, achieving intelligent and precise realization of grassland animal detection and tracking has important research significance and practical value. Figure 3-1 shows images of grassland animals taken by UAV at low altitudes. It can be seen that with increased UAV height, the proportions of targets in the picture become smaller and smaller; therefore, it is necessary to improve the ability of models to detect small objects when detecting normal objects. If a drone is flying at high altitude, this presents a huge challenge in detection. At present, there are many algorithmic models that are able to detect wild animals, such as the algorithm proposed by Mateusz Choinski et al.[129] for monitoring the number of wild animals and the algorithm proposed by Dario G. Lema et al.[130] for detecting whether livestock activities exist in specific terrains, but the performance of these algorithms in real-time needs to be improved.

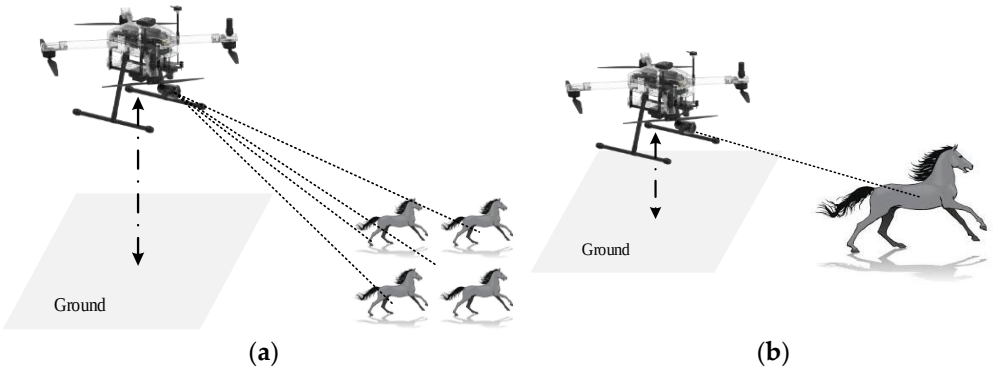


Figure 3-1: (a) The targets are small in the image; (b) The targets are large in the image.

At present, object detection in images can be roughly divided into two categories: the first includes one-stage detection methods, such as YOLO and SSD[35,34,36,38]. The other includes two-stage detection methods, the most representative of which is Faster RCNN[29]. The reasoning speed of the one-stage detection method is relatively high. The difference is that the two-stage detection method has higher positioning and target recognition accuracy, while the reasoning speed is relatively low.

In this chapter, a lightweight grassland animal object detection system is designed based on Yolov5. First, Squeeze-and-Excitation Networks are introduced to improve the expressiveness of the network model. Specifically, the importance of each channel is automatically obtained by learning, and then features that are useful are promoted and features that are of little use to the task at hand are suppressed according to this level of importance. Secondly, considering the redundancy of feature

map channels, the convolutional layer of branch 2 in the BottleNeckCSP structure is deleted, and 3/4 of its input channels are directly merged with the results of branch 1 processing, so that the number of 1×1 convolutional layer channels is reduced, which reduces the number of model parameters with guaranteed accuracy. Next, in the SPP module of the network model, a 3×3 maximum pooling layer is added to improve the receptive field of the model and thus the detection of small targets. Finally, the trained model was applied to NVIDIA-TX2 with an FPS of about 26.

3.2 Related Work

In this section, previous works related to the proposed method are reviewed. At present, object detection technology is used in many fields in combination with object detection, such as in forest fire detection[131], identification of insulator defects on pylons[132], and aerial vehicle detection[133]. At the same time, there have been many studies on object detection for wildlife detection, such as O-YOLOv2, YOLOv2[134], YOLOv3, Tiny-YOLOv3[135], YOLOv4-uw[136], Faster R-CNN, Modified Faster R-CNN, RetinaNet[137], CenterNet, improved CenterNet[138], and other models, the performances of which are shown in Table 3-1. Although many models have high detection accuracy, the large scale of the models and the large number of parameters leads to their ability to perform real-time detection in application being insufficient. Jinbang Peng et al.[137] used Faster R-CNN and modified Faster R-CNN models, respectively, to detect wild animals. Although the detection accuracy was high, the detection speed was very low. The detection speed of the Faster R-CNN model was 3 fps, and the detection speed of the Modified Faster R-CNN model was 2 fps.

With the advancement of technology, the application of UAVs is everywhere in daily life, and research based on UAV vision object detection is common. The current application is more based on the detection of pedestrians and vehicles by drones[139,140,141]. The SlimYOLOv3 model proposed by Pengyi Zhang et al.[142] not only has a high detection accuracy but also meets the practical needs of UAVs in real-time. Yuanyuan Hu et al.[143] applied the object detection model to UAV countermeasures, which is a new research direction based on UAV object detection and also achieved good results in terms of real-time and accuracy. Small target detection based on UAV vision is also a research hotspot. The UAV-YOLO model proposed by Mingjie Liu et al.[144] improves the accuracy of small target detection by adding spatial information. Haijun Zhang et al.[145] provide a multi-scale dataset based on UAV vision, named MOHR, and this dataset is of great significance for monitoring in the industry.

The purpose of this chapter is to design a lightweight real-time object detector that can be deployed to an embedded platform and better integrated with UAVs. At the same time, the detector should accommodate as much as possible the change in altitude of the UAV during actual flight.

3.3 Materials and Methods

3.3.1 YOLOv5 Network Model

The YOLO model has always been widely used. There have been five updated versions, from YOLOv1 to YOLOv5. With continuous improvement and innovation, it

Table 3-1: Performance comparison of different regression loss functions.

Object Detection Networks	**Precision**	**Recall**	**mAP**	**Average DetectionSpeed (s/pic)**	**Reference**
O-YOLOv2	0.94	0.94	0.94	0.17	134
YOLOv2	0.91	0.88	0.87	0.17	134
YOLOv3	–	0.64	0.825	0.25	135
Tiny-YOLOv3	–	0.49	0.6241	0.068	135
YOLOv4-uw	–	–	0.7534	0.023	136
Faster R-CNN	0.82	0.88	–	0.32	137
Modified Faster R-CNN	0.92	0.96	–	0.55	137
RetinaNet	0.81	0.97	–	0.11	137
CenterNet	94.3	94.9	0.8924	0.032	138
improved CenterNet	96.8	95.5	0.9361	0.027	138

has been used by deep learning enthusiasts as one of the preferred frameworks for object detection[146,147]. The official code of YOLOv5[148] provides a total of five versions of the object detection network: YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and YOLOv5n. YOLOv5n is mainly designed for mobile and CPU environments; it is fast, but not accurate. Among the other four versions, YOLOv5s is the network with the narrowest feature map width and the shallowest depth. The following three models continue to widen and deepen these aspects, respectively. The YOLO network model is mainly composed of the backbone, neck, and prediction layers. The backbone is a convolutional neural network that aggregates different image granularities and simultaneously forms image features[149,150]. The neck is a series of network layers that mix and combine image features. Its main function is to transfer image features to the prediction layer. The prediction layer predicts the features of the image, generates the bounding box of the detection target, and predicts the type of the target object[151,152].

Backbone Module

The first layer of the backbone is focus. The main function of this module is to enrich the training dataset; in particular, random scaling is used to increase the number of small targets in the training process, improving the robustness of the network model, and greatly improving its ability to detect small targets.

The default input of YOLOv5s is $640 \times 640 \times 3$, and the focus layer copies it into four, and then cuts the four pictures into four $320 \times 320 \times 3$ slices using a slicing operation. Then, the four slices are stitched together depth wise, making the output $320 \times 320 \times 12$, before being passed through a convolutional layer with a number of convolution kernels equal to 32 in order to generate a $320 \times 320 \times 32$ output. Finally, the batch normalization and activation function are applied, and the results are used as input to the next convolutional layer.

BottleNeckCSP is in the third layer of the backbone, and is divided into two main parts, BottleNeck and CSPNet[153]. BottleNeck is a classic residual network structure. The first is a 1×1 convolutional layer (conv+batch_norm+leaky relu), the next is a 3×3 convolutional layer, and finally, the initial input is added through the residual network structure. The full name of CSPNet is Cross Stage Partial Network, and it solves the problem of repeated gradients in other large convolutional network structures[154,155,156].

Neck Module

The main function of the neck module is to generate a feature pyramid and transfer the features of the image to the prediction layer. The feature pyramid can be used to optimize the network model's detection of target objects of different scales, and then to identify the same target objects at different sizes and scales. Before the PANet[157] structure came out, FPN was always the preferred structure for the feature aggregation layer of the object detection framework. In the research on YOLOv4, it has been found that the most suitable feature fusion network for YOLO is PANet. Therefore, both YOLOv4 and YOLOv5 use PANet as the neck to aggregate features.

PANet is based on the Mask R-CNN and FPN frameworks, and on this basis,

the dissemination of information is optimized[158,159]. The feature extractor of the network uses a bottom-up path FPN structure, thereby optimizing the propagation of low-level features. The feature map of the previous stage is used as the input of each stage of the third path, and a 3 × 3 convolutional layer is applied to process it at the same time. The output is added to the feature map of the same stage of the top-down path through the horizontal connection, and these feature maps provide information for the next stage. At the same time, adaptive feature pooling is used to restore the damaged information paths between all feature levels and each candidate area and aggregate each candidate area on each feature level in order to prevent arbitrary allocation[160,161].

Prediction Module

The prediction module performs the final detection, and an anchor box is applied to the output feature map, generating an output vector with category probability, confidence score, and bounding box. On the anchor, YOLOv5 uses cross-grid matching rules to distinguish the positive and negative samples of the anchor. The loss function uses GIOU_loss, and the confidence loss and category loss use the binary cross-entropy loss function.

Pre-Training

At this stage, it is very difficult to obtain large datasets when users have to take pictures themselves. At the same time, if the dataset is too small, overfitting will occur when training the model, which will lead to the model having poor generalization ability and robustness. Therefore, users typically do not train network models from scratch for a given item. The amount of data in this experiment was also limited, and the training results are likely to exhibit overfitting. To solve this problem, we adopted the transfer learning method to improve model generalization[162]. We used the backbone of the COCO dataset to pre-train the network model and used the trained backbone to train the wildlife dataset. This method reduced the size of the training dataset, increased the training speed of the model, and effectively solved the problem of model overfitting. Since transfer learning allows the model to learn using different types of data, it is better at capturing the internal connections of the problem to be solved.

3.3.2 Improved YOLOv5

The improved YOLOv5s network model is shown in Figure 3-2. To improve the performance of the model, SENet network is added after the first three BottleneckCSP and the BottleneckCSP in the three detection branches. At the same time, in order to reduce the amount of parameters, the convolution of branch 2 in BottleNeckCSP structure is deleted, and 3/4 of its input channels are directly merged with the results of branch 1 processing. Finally, in order to improve the ability of the model to detect small targets, a 3 × 3 max-pooling layer is added to the SPP module to improve the receptive field of the model.

Addition of the SENet Network Structure

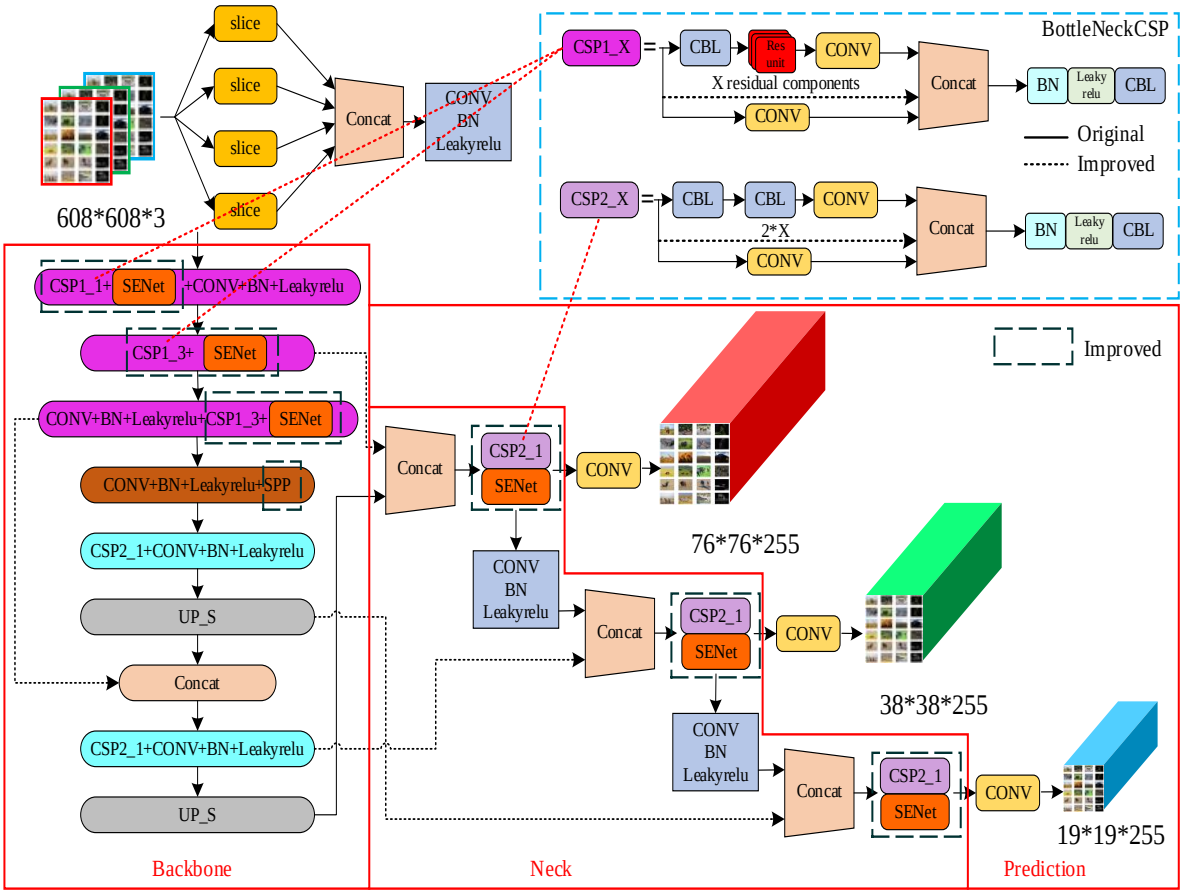


Figure 3-2: The network structure of improved YOLOv5s.

Since the shape and appearance of grassland animals are different from the background color in the image, in order to improve the detection accuracy for grassland animal targets[163,164,164,165], the SENet network is introduced[166], the structure of which is shown in Figure 3-3.

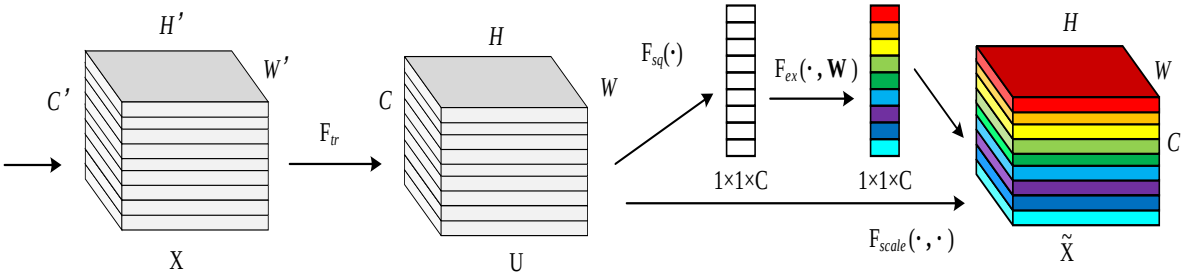


Figure 3-3: Squeeze and excitation module structure (Copyright IEEE, 2022.).

First, the F_{tr} step is a conversion operation. In fact, it is a standard convolution operation in the structure, and the input and output are defined as: $F_{tr} : X \rightarrow U, X \in R^{H' \times W' \times C'}, U \in R^{H \times W \times C}$. The specific form of this F_{tr} is shown in Equation 3-1, where V_c represents the c-th convolution kernel, and X^s represents the s-th input.

$$u_c = v_c * X = \sum_{s=1}^{c'} v_c^s * X^s \tag{3-1}$$

The U obtained by F_{tr} is the second three-dimensional matrix in the structure diagram, and u_c represents the c-th two-dimensional matrix in U. What follows is the squeeze operation, the specific form of which is shown in Equation 3-2. In fact, squeeze converts the $H \times W \times C$ input into $1 \times 1 \times C$ a output.

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j) \tag{3-2}$$

Next is the excitation operation, the specific form of which is shown in Equation 3-3. The result obtained by squeeze, above, is z. First, multiply W_1 by z. The dimension of W_1 is $\frac{C}{r} \times C$, and r is the scaling parameter. Its function is to reduce the number of channels, thereby reducing the amount of calculation required. In addition, because the dimension of z is $1 \times 1 \times C$, the dimension of $W_1 z$ is $1 \times 1 \times C/r$; then, through the ReLU layer, the dimension remains unchanged. Then multiply by W_2; the dimension of W_2 is $C \times C/r$, so the output dimension is $1 \times 1 \times C$, and finally through the sigmoid function, s is obtained.

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{3-3}$$

It can be seen from the above that the dimension of s is $1 \times 1 \times C$, and s is used to describe the weight of the feature map C in U. After obtaining s, it is possible to operate on the original U. The specific form is as shown in Formula 3-4, where s_c represents the weight. Therefore, it is equivalent to multiplying each value in the u_c matrix by s_c, which corresponds to F_{scale} in Figure 3-3.

$$\widetilde{x}_c = F_{scale}(u_c, s_c) = s_c u_c \tag{3-4}$$

The core idea of SENet is to learn the target feature weight through the loss function, and by improving the effective feature map weight. Train the network model by reducing the weight of the feature map that is invalid or has a small effect, so as to achieve better results. The SENet network structure requires a small amount of calculation, while at the same time effectively improving the expression ability of the network model and optimizing it. Therefore, the SENet network is embedded in the YOLOv5s model to improve the detection accuracy of the model, as shown in Figure 3-4.

Figure 3-4: Optimized CSP1_X and CSP2_X module.

After adding the SENet module, the number of parameters of the model increased by about 3 percentage points, and the running speed was basically the same as that of the original network. Meanwhile, in order to reduce the number of parameters of the model, the weight parameter of the model channel was changed from 0.5 to 0.45 under the condition of ensuring the accuracy.

Improve BottleNeckCSP Module

Because it is necessary not only for the UAV object detection algorithm to accurately identify animals in different environments in the grassland, but also to reduce the model as much as possible and increase the calculation speed in order to realize real-time detection using a UAV, the BottleNeckCSP structure in the backbone network of the YOLOv5s framework is optimized. This ensures that, while improving the detection speed, the accuracy of object detection does not change significantly, thereby resulting in a lightweight UAV object detection model.

According to the architecture of the YOLOv5s network model, the backbone network contains three BottleNeckCSP modules, and there are more convolutional layers in this module. Although the convolutional layer can be used to effectively extract the features of a picture, there are also more parameters in the convolutional layer, which means that there are more parameters in the model, which leads to a decrease in calculation speed. In response to this problem, the BottleNeckCSP module is optimized in the backbone network. The convolutional layer of branch two is deleted, and the input of the BottleNeckCSP module is merged directly with the result of the branch one processing. This will lead to the increase of feature map channels after concat, so that the parameters of convolution will increase in output, and the number of parameters will remain unchanged after calculation. Considering the redundancy of the feature graph, a layer was deleted every four channels in the input channel of branch 2 to make the input channel 3/4 of the original, so as to reduce the number of parameters of the model under the condition of ensuring accuracy. The structure is shown in Figure 3-5(a), (b).

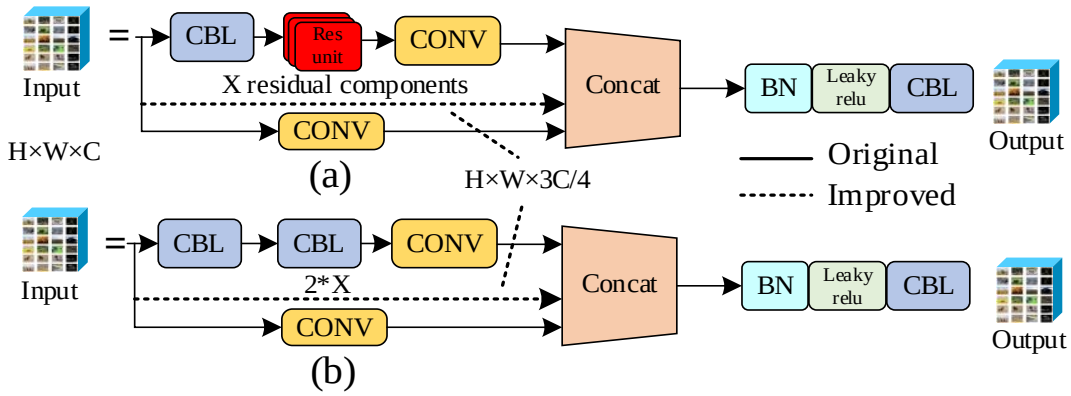


Figure 3-5: (a) The network structure of improved BottleNeckCSP_1 module; (b) The network structure of improved BottleNeckCSP_2 module.

Optimize the SPP Module

While the drone is performing aerial photography, if the altitude is too high, it will cause the target to have small proportions in the image. The size of the input feature map of the SPP module is $512 \times 19 \times 19$. After the convolution kernel of $256 \times 512 \times 1 \times 1$, the number of channels of the feature map changes, and the size of the output feature map is $256 \times 19 \times 19$. Then, self-sampling this feature map with three parallel max-pooling layers, and then splicing the output feature map

into the channel, outputting a feature map with a size of 1024 × 19 × 19. Finally, a feature graph with an output size of 512 × 19 × 19 is obtained after the 512 × 1024 × 1 × 1 convolution kernel. To improve the detection accuracy of small and medium targets, a 3 × 3 maximum pooling layer is added to the SPP module to improve the receptive field of the model. At the same time, in order to ensure that the number of input channels of the CSP2_1 module is consistent with the number of output channels of the SPP module, the weight matrix of the second convolution kernel in the SPP module is then increased by 1/4 of the number of channels. The improved SPP module is shown in Figure 3-6.

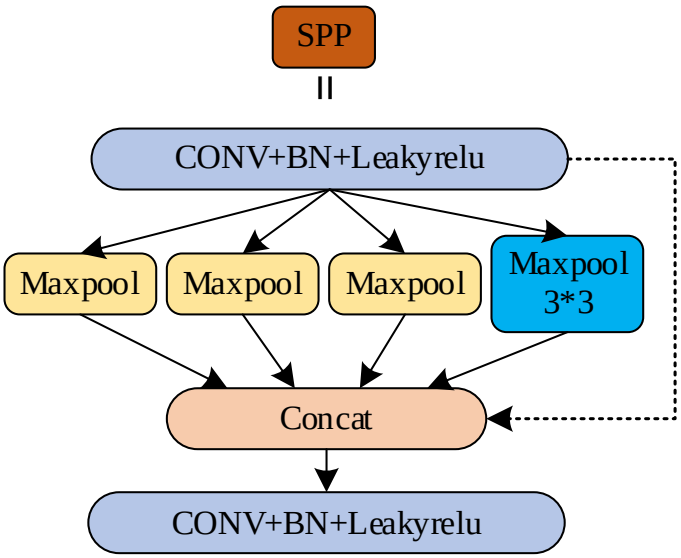


Figure 3-6: The network structure of improved SPP module.

Other Tricks

The YOLOv5s model has three detection feature maps, which are obtained from 8, 16 and 32 times of down-sampling respectively. The feature maps are 76 × 76, 38 × 38 and 19 × 19, respectively. The small feature map is used to detect the large target, and the large feature map is used to detect the small target. This chapter tried to replace feature maps of different depths for splicing, so that the feature map paid more attention to the size of the target in the data set. However, due to the height change of UAV, the proportion of the target in the image changed greatly, so the experimental results were not ideal. Finally, the original network splicing method is adopted.

Anchor boxes of different sizes and proportions are set for feature maps of different sizes in YOLOv5s model. These anchor boxes are used to frame the target object. Through labeling, it can be found that the ratio of label width and height in the data set of this experiment is roughly distributed at 1:2 and 2:1. Therefore, it is necessary to modify the size of anchor boxes according to its own data characteristics before training. In this experiment, the size of anchor boxes 33 × 23 in 78 × 78 feature map was changed to 33 × 16, and the size of anchor boxes 116 × 90 and 373 × 326 in 19 × 19 feature map was changed to 116 × 60 and 350 × 180,

respectively. The size of other anchor boxes basically conforms to the ratio of label width to height, so no modification will be made.

3.4 Results

3.4.1 Experimental Setup and Results Analysis

Dataset Introduction

Part of the dataset is generated by image-downloader, an open-source project that allows users to download images from Google, Bing, and Baidu websites by entering the name of the Image[167]. The other part of the data set mainly comes from Vision China, and this website has video data specifically for aerial photography[168]. The dataset includes six prairie animals, elephants, zebras, bison, wild horses, giraffes, and hippos, each with about 500 images[169]. Consideration of different time periods, different angles, different distances and occlusions, etc., was achieved by rotating the pictures at different angles, adjusting the contrast, etc. The number of datasets was thus increased to 4 times the original number. The makesense.ai tool was used to label the grassland animals in the picture and divide the dataset into a training set and a test set at a ratio of 9:1. There were 3000 images in the basic dataset, and the resolution of most of the images was 1200 × 960. After data amplification, the total dataset contained 12,000 images. Meanwhile, YOLOv5 uses many effective data processing methods to increase the accuracy of the training model and reduce the training time. The main methods of data amplification are Mosaic and Cutout. In addition to these two methods, YOLOv5 also uses image perturbation, changes in brightness, saturation, and hue, the addition of noise, random scaling, random cropping, flipping, rotating, random erasure, etc., to expand the amount of data.

Model Training

In this experiment, Indexes such as Precision, Recall, F1, AP, mAP_0.5 and mAP_0.5:0.95 were selected to evaluate the performance of the grassland animal object detection model after training.

Precision reflects the ability of the model or classifier to correctly predict the accuracy of positive samples. The larger the value, the better the performance. Recall is the proportion of positive samples predicted to be positive samples to the total positive samples, and its performance is the same as Precision. Precision and Recall influence each other. Generally, if the accuracy rate is high, the recall rate will be low, and if the accuracy rate is low, the recall rate will be high. The $F1$ value is the weighted harmonic average of precision and recall. Taking an elephant to be detected in the picture as an example, TP means that the target in the picture was correctly recognized as an elephant, FP means that another target was detected was incorrectly recognized as an elephant, and FN means that the target in the picture was wrongly identified as belonging to another category.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3-5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3-6}$$

$$F1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{3-7}$$

AP represents the area under the Precision–Recall curve, while mAP denotes mean average precision, which is the average value of each category of AP. mAP_0.5 refers to the average value of all APs when the IOU threshold is set to 0.5. mAP_0.5:0.95 represents the average mAP for different IOU thresholds (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95). C represents the number of target types, N represents the number of IOU thresholds, Krepresents the current IOU threshold, $P(K)$ and $R(K)$ represent precision and recall.

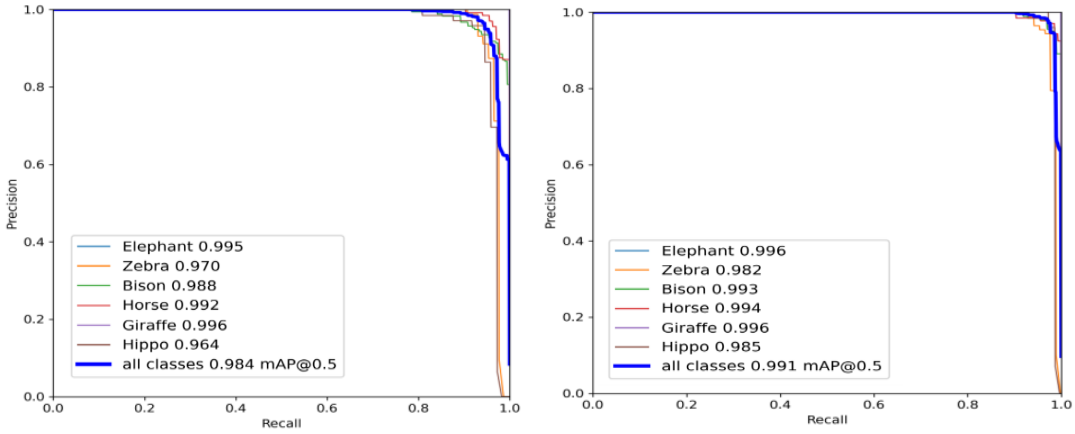
$$\text{AP} = \sum_{k=1}^{N} P(K)\Delta R(K) \tag{3-8}$$

$$\text{mAP} = \frac{1}{C}\sum_{k=1}^{N} P(K)\Delta R(K) \tag{3-9}$$

$$\Delta R(K) = R(K) - R\,(K-1) \tag{3-10}$$

Model Comparison

The PR curves of the YOLOv5s model and the improved YOLOv5s model after training are shown in Figure 3-7.



(a) **(b)**

Figure 3-7: (a) YOLOv5s PR curve; (b) improved YOLOv5s PR curve.

The PR curves for each class in different models are presented in Figure 3-7, and the specific information is summarized as shown in Table 3-2. In the improved YOLOv5s model, only the average accuracy of giraffe detection was not improved, and the average accuracy of detection of the other five grassland animals was improved. It can be seen that the overall performance of the improved YOLOv5s model was better than that of the original model.

Table 3-2: Average Precision (IOU = 0.5) obtained for each evaluated object detection algorithm.

Class	YOLOv3(AP)	YOLOv5s(AP)	Improved YOLOv5s(AP)
Elephant	0.923	0.995	0.996
Zebra	0.706	0.970	0.982
Bison	0.942	0.988	0.993
Horse	0.805	0.992	0.994
Giraffe	0.783	0.996	0.996
Hippo	0.812	0.964	0.985

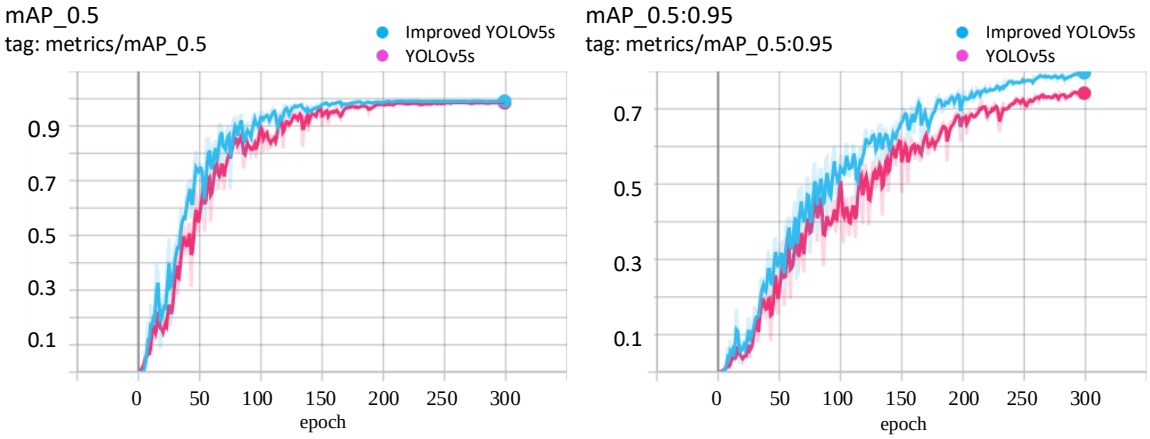


Figure 3-8: Comparison chart of mAP_0.5 and mAP_0.5:0.95.

Figure 3-9: Comparison between Precision and Recall.

During the training process, tensorboard was used to draw the relevant curve. The data curves of Precision, Recall, mAP_0.5 and mAP_0.5:0.9 are shown in Figure 3-8, 3-9. The blue color corresponds to the improved YOLOv5s data curve, and the pink color represents the YOLOv5s data curve. In terms of speed and accuracy, the improved YOLOv5s model is better.

Loss Function Comparison

The last layer of the network model was compared with the objective function to obtain the loss function, the error update value was calculated, and the first layer was reached layer by layer through backpropagation, and the ownership value was updated together at the end of the backpropagation. The loss function can more intuitively reflect the performance of a classifier or model. The smaller the loss, the better the performance of the model or classifier. As shown in Figure 3-10, the data curves of box_loss, cls_loss and obj_loss of the two models are shown in the figure. The blue color corresponds to the improved YOLOv5s data curve, and the pink color represents the YOLOv5s data curve. It can be seen that with continuous training, the performance of the two models improved gradually, and the improved YOLOv5s model converges relatively quickly.

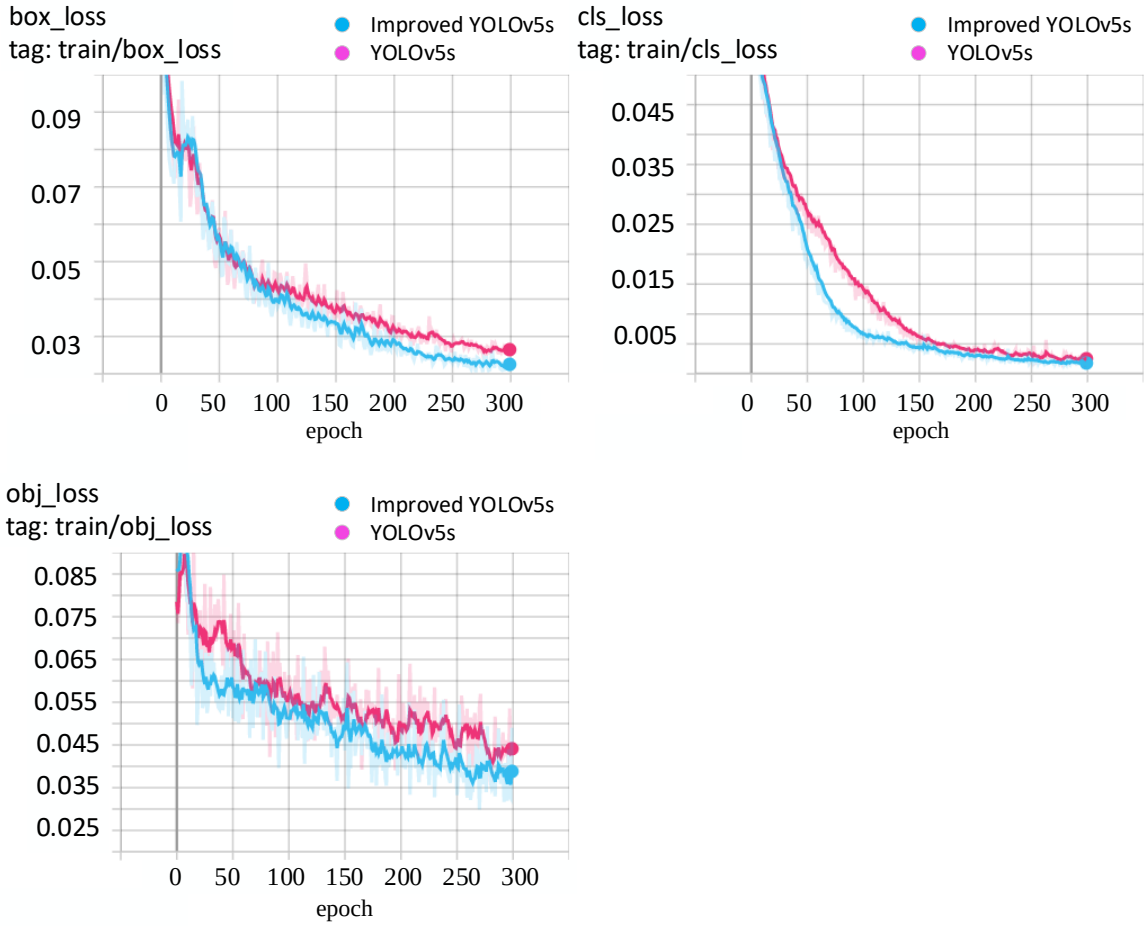


Figure 3-10: The loss function comparison diagram includes box_loss, cls_loss and obj_loss.

3.4.2 Test Results and Analysis

The results of the detection tests are presented here. The test devices are the same as for the training machine. However, only in 3.4.2, the test device is an NVIDIA-TX2 and the OS is Linux. This is to verify the computation speed and memory usage in an experimental environment similar to that of an actual UAV.

Test Result

The test set was used to verify the improved YOLOv5s model, and its actual effect is shown in Figure 3-11. It can be seen from the figure that the improved YOLOv5s model was able to correctly identify the six grassland animals in different time periods, from different perspectives, and with different target proportions.

Comparison of Results

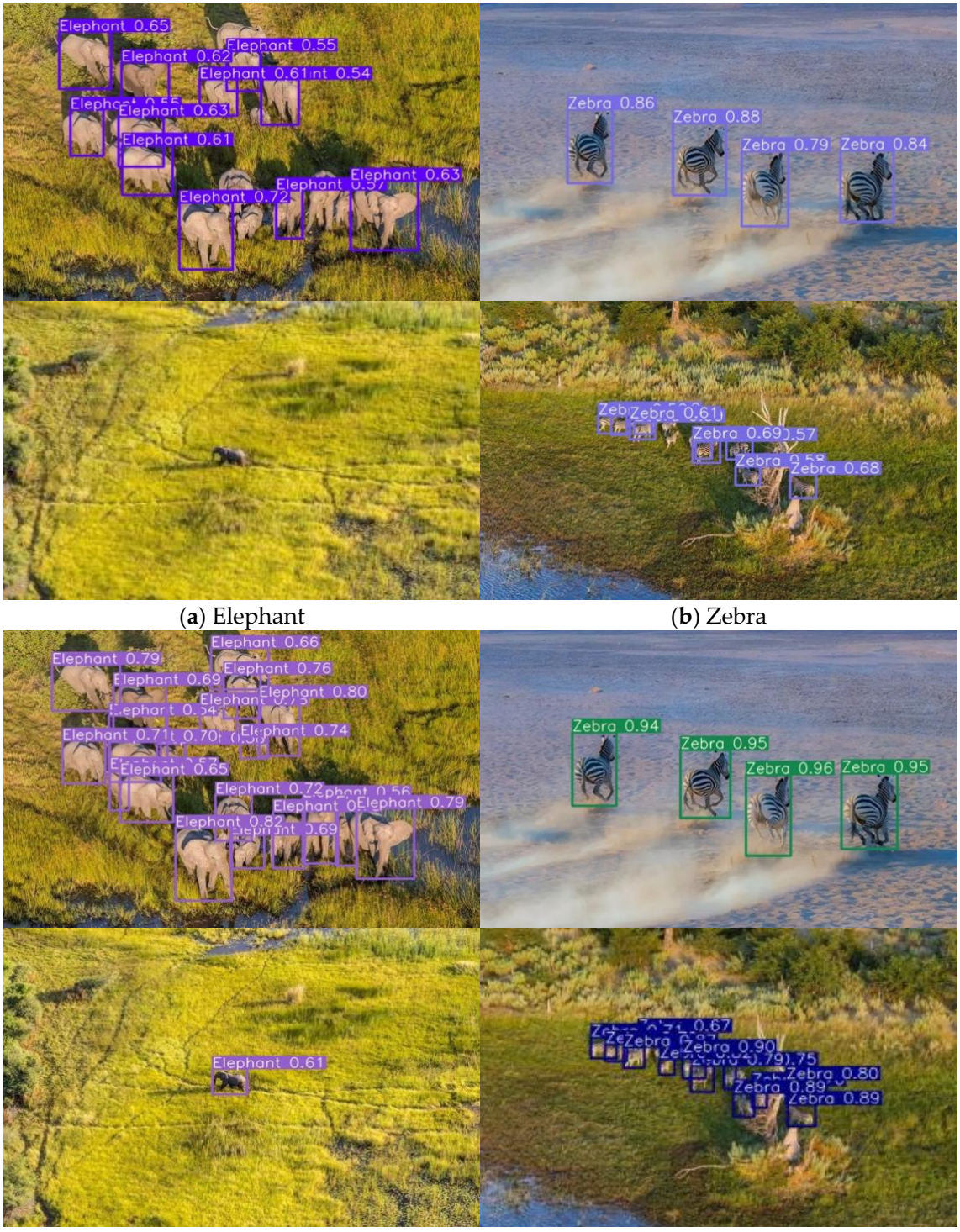
A comparison of the actual application of the improved YOLOv5s model and the original YOLOv5s model is shown in Figure 3-12, 3-13, with the value of IOU set to 0.5. If the value is lower than 0.5, the detection box will not be displayed. In Figure 3-12(a), (c), the original YOLOv5s model was not able to identify elephants that were relatively small in the picture; elephants with moderate proportions in the picture could not be identified completely, only a part of them can be identified. The improved YOLOv5s model can correctly identify it. In Figure 3-12(b), (d), the original YOLOv5s model was not able to identify zebras that were relatively small in the picture. The improved YOLOv5s model was able to correctly identify it. In Figure 3-13(a), (c), the improved YOLOv5s model was also able to correctly identify small and medium targets. In Figure 3-13(b), (d), both the original YOLOv5s model and the improved YOLOv5s model were able to achieve a correct recognition, but the improved YOLOv5s model had better recognition accuracy than the original YOLOv5s model.

Performance Comparison with Other Networks

To further verify the performance of the improved model at detecting grassland animals, the improved YOLOv5s model was compared with other models in the test set. mAP_0.5, mAP_0.5:0.9 and average detection speed were taken as the evaluation indicators of the model, and a comparison of the results is shown in Table 3-3. The test device was NVIDIA-TX2.

It can be seen from Table 3-3 that the mAP_0.5 and mAP_0.5:0.9 values of the improved YOLOv5s model are the highest, indicating that the performance of the improved YOLOv5s was the best among the YOLOv3, EfficientDet-D0, YOLOv4, YOLOv5s and improved YOLOv5s models. As far as the detection speed of the network model is concerned, the improved YOLOv5s model has an average detection speed of 26 fps in NVIDIA-TX2, which is a bit slower than the initial YOLOv5s model, but is better than the YOLOv3 model, EfficientDet-D0 model

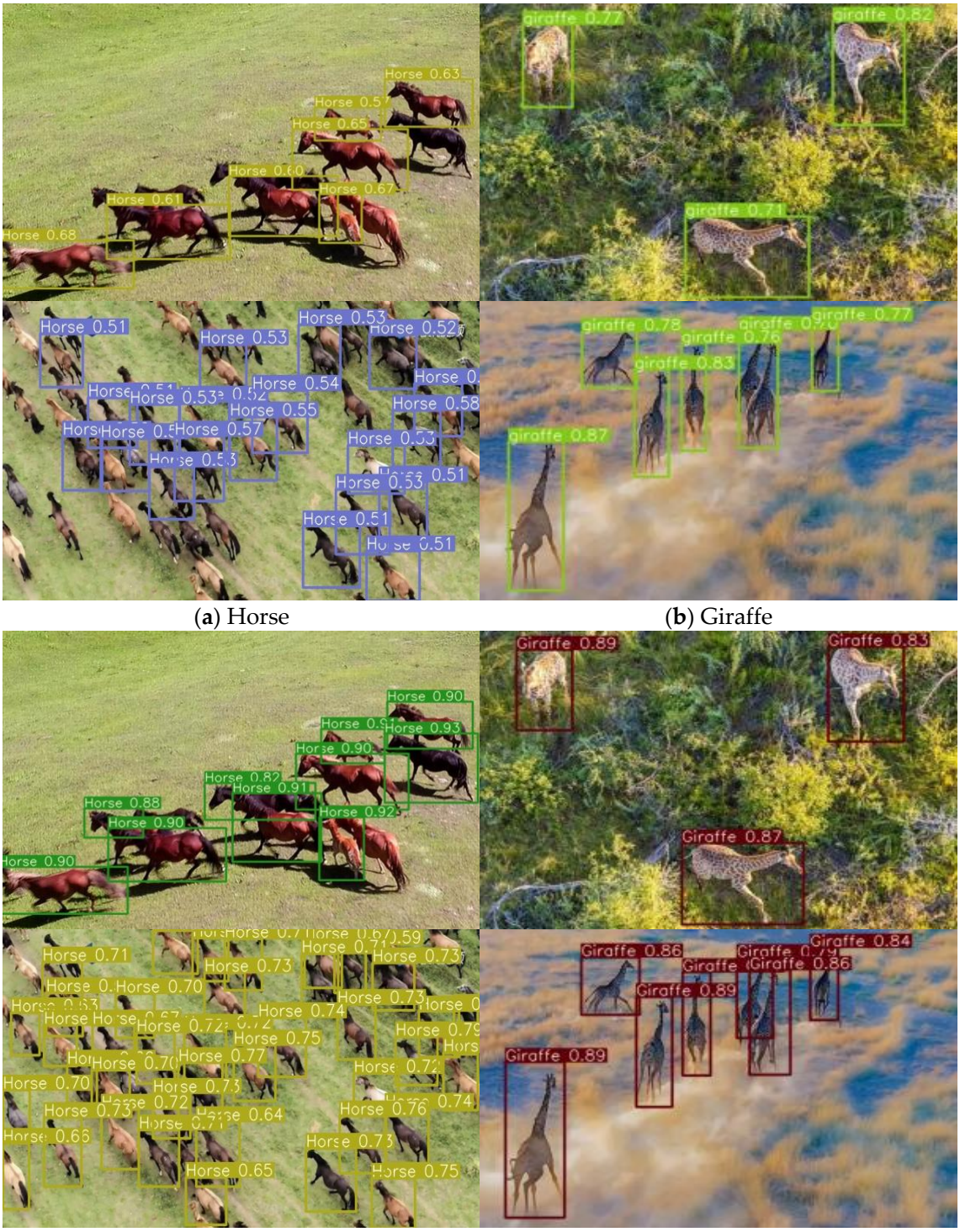
Figure 3-11: Grassland animal test results.



(**a**) Elephant (**b**) Zebra

(**c**) Elephant (**d**) Zebra

Figure 3-12: (a) Test results for elephant with the original YOLOv5s model; (b) test results for zebra with the original YOLOv5s model; (c) test results for elephant with the improved YOLOv5s model; (d) test results for zebra with the improved YOLOv5s model.



(**a**) Horse (**b**) Giraffe

(**c**) Horse (**d**) Giraffe

Figure 3-13: (a) Test results for horse with the original YOLOv5s model; (b) test results for giraffe with the original YOLOv5s model; (c) test results for horse with the improved YOLOv5s model; (d) test results for giraffe with the improved YOLOv5s model.

Table 3-3: Performance comparison of four object detection networks.

Object Detection Networks	**mAP_0.5**	**mAP_0.5:0.95**	**Average Detection Speed (s/pic) (TX2)**	**Size of Model (MB)**	**Number of Parameters**
YOLOv3	0.786	0.526	0.313	235	6.15×10^7
EfficientDet-D0	0.942	0.676	0.091	15	3.83×10^6
YOLOv4	0.965	0.708	0.051	244	6.39×10^7
YOLOv5s	0.961	0.691	0.033	14	7.25×10^6
Improved YOLOv5s	**0.972**	**0.742**	**0.039**	**12.8**	$\mathbf{6.62 \times 10^6}$

and the YOLOv4 model, which meets the requirements of drones for real-time detection of grassland animals. At the same time, it can be seen from Table 3-3 that the size of the improved YOLOv5s model is only 12.8 MB, which is smaller than the other models. Experiments have proved that the improved YOLOv5s model not only ensures the accuracy of object detection, but also ensures that the network model is lightweight. In summary, among the four network models proposed in Table 3-3, the improved YOLOv5s model has the highest mAP_0.5 value and mAP_0.5:0.9 value, and the scale of the model is also relatively small. At the same time, the detection speed is also better than that of the YOLOv3 model, EfficientDet-D0 model and the YOLOv4 model. Although the detection speed is lower than that of the initial YOLOv5s model, it can meet the needs of real-time detection using UAVs.

Pascal Voc 2012 Dataset Validation

The public dataset selected for this experiment is Pascal voc 2012, with 20 category types. Its tag format is xml, but YOLOv5 needs txt format file, so we need to convert the xml format tag to txt format first. Then the 17,125 images were divided into training and validation sets, with 13,637 images in the training set and 3488 images in the validation set. The training conditions are consistent with those described above, and their results on the validation set are shown in Figure 3-14. In the mAP_0.5 and mAP_0.5:0.95 metrics, the improved YOLOv5 is 0.047 and 0.05 higher than the original model, respectively.

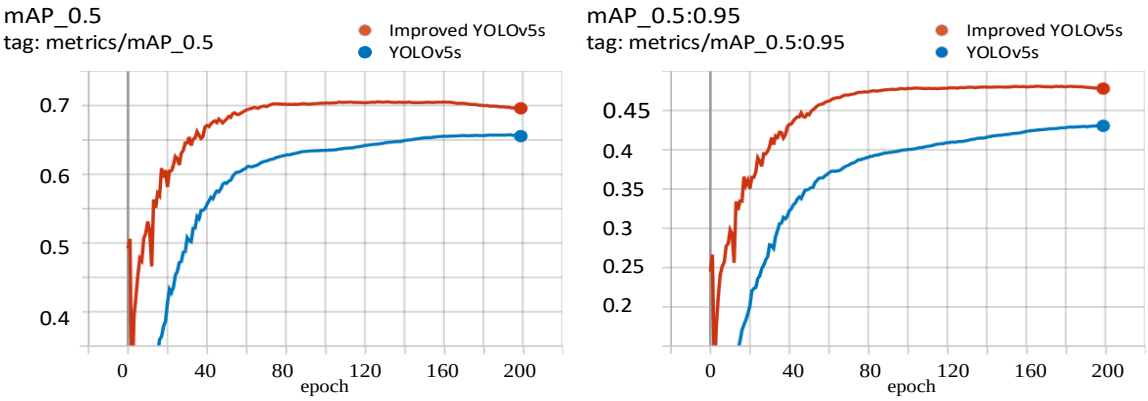


Figure 3-14: Pascal voc 2012 dataset validation comparison.

3.5 Discussion

For the problem of false positives, test sets were selected to test the performance of the model with 1612 label targets. Positive samples with an IOU threshold greater than 0.5 and negative samples with an IOU threshold less than 0.5 were selected. The number of true positives was 1548, the number of false positives was 48, and the number of false negatives was 64. False positive image types tend to have similarities between the target to be detected and the background, which may be mainly due to the following three reasons. Firstly, the content of the image. When training the model, in order to improve the generalization of the model, data enhancement is generally used to simulate complex situations such as different

illumination and different angles in the image. This process may make some images too bright or too dark. After these images have been extracted by the model, if they are similar to some background features extracted by the model, the model will detect the background as an object. Secondly, it is necessary to consider the scope of the bounding box. In the process of model training, it is necessary to provide the position of the target in the picture, that is, the enclosing rectangle. However, the general target to be detected is not a rectangle, and there will be some background contents inside the label, which could also be responsible for the false positive. Thirdly, when the drone is flying at high altitudes, the characteristics of wild animals are proportionally relatively close, which could also be a reason for false detection.

A test video can be found at reference[170]. The overall effect is ok, but there are some shortcomings. Some elephants are sometimes misidentified as giraffes. We guess that the reason for this is that during the training of the model, part of the data enhancement darkened the picture to which the giraffe belongs, which would make its features similar to those of the elephant in the video, thus leading to misidentification. The problems of the missed detection of small targets and the difficulty of detecting occluded objects in videos still need further research.

One thing to note is that the lightness of the model facilitates deployment. The speed of wildlife detection is also extremely important for drones. The recognition speed in Jinbang Peng's[137] paper was 2–3 fps, and was not able to meet the requirements of real-time detection by UAV. The detection speed of the model proposed in this chapter is 26 fps, 12 times of its detection speed, which meets the requirements of real-time detection using UAVs. Although the YOLOV4[36] model has high detection accuracy, its model is too large, which is not conducive to the deployment of embedded devices. The model YOLOv4-uw proposed by Chen L et al.[136]. has reached a detection speed of 43 fps, but its accuracy is relatively low, which can easily cause the phenomenon of missed or false detection. The application of the Tiny-YOLOv3 model by Adami D et al.[135]. meets the requirements of lightweight deployment, but its observation of animals mainly from the ground perspective does not meet the needs of this chapter applied to UAVs. In conclusion, the target detection model proposed in this chapter takes into account the accuracy and real-time requirements. The accuracy of detection is ensured while real-time detection is performed. At the same time, this chapter solves to a certain extent the problem that the change of target occupancy ratio makes detection difficult.

In this chapter, the selection and design of the model were mainly carried out considering actual application, where the model can be easily deployed using embedded devices, in order to achieve real-time object detection. The characteristics of light weight and fast detection make the YOLOV5s model highly competitive in a variety of embedded device deployments. In conclusion, the model proposed in this chapter has the following advantages. Firstly, the model can automatically detect wildlife in the video stream. Secondly, the improved YOLOV5s model is very small in scale, which makes it easy to deploy to a variety of embedded devices. This reduces hardware costs for users, which is of great value in practical applications. Thirdly, the detection speed of the improved YOLOV5s model is very fast, easily meeting the needs of real-time detection of wild animals. However, most of the dataset in this chapter is in relatively good light, with a small number of dusk and night images. Therefore, working at night may not be applicable to the model

proposed in this chapter. At the same time, if the UAV flight is high and the proportions of the target are small, the target will be difficult to detect, which is the disadvantage of the model proposed in this chapter.

3.6 Conclusion and Future Work

To realize real-time detection of grassland animals using aerial drones, this chapter proposes a real-time detection method for grassland animals based on the YOLOv5 network model. In the improved YOLOv5s model, in order to improve the accuracy of object detection, a SENet structure is added. To achieve a lightweight model, the BottleneckCSP module in the Neck layer was replaced with the BottleneckCSPS_X module. To realize the detection of small and medium grassland animal text, the SPP module is optimized and a 3×3 maximum pooling layer is added to improve the receptive field of the model. The experimental results show that compared with YOLOv3, EfficientDet-D0, YOLOv4, and YOLOv5s, the improved YOLOv5s network model demonstrated an increase of 0.186, 0.03, 0.007, and 0.011 in the value of mAP_0.5, an increase of 0.216, 0.066, 0.034 and 0.051 in the value of mAP_0.5:0.95, and an average detection speed of 26 fps. At the same time, the scale of the improved model is also small and meets the needs of aerial drones for the real-time detection of grassland animals.

To address the limitations of the model proposed in this chapter, a searchlight could be hung on the drone to facilitate the collection of pictures of wild animals at night. Add the collected pictures to the training set to solve the problem of observing the habits of wild animals at night. At the same time, in practical applications, observing the living habits of wild animals requires tracking and observing the target. The model proposed in this chapter can be fused with the model of object tracking. The fused model can get the position information of the target more stably, transmit this information to the UAS, and use coordinate conversion to get the 3D information of the target. According to this information, the target can be tracked easily by using UAV control technology. In order to cope with some dead ends in tracking, the camera angle can be controlled by using a servo, which can greatly improve the stability of tracking. Because the drone is too high, the target proportion is small, so that the target is difficult to detect is also a problem to be solved. In addition, it would also a good research direction to deploy the model proposed in this chapter in other embedded devices for application in the field of robotics.

4 UAV Autonomous Inspection System for High-Voltage Power Transmission Line

4.1 Introduction

The stable transmission of electricity by high-voltage lines is of great importance to modern industry and people's lives[171,172,173,174]. In daily life, power departments at all levels should carry out daily maintenance of high-voltage lines to prevent damage to them by lawless elements or by bad weather, natural losses, etc. The traditional high-voltage line inspection approach is walking along the line or with the help of transportation, while using binoculars and infrared thermal imaging cameras, such as line equipment and channel environment, for proximity inspection and detection, which are low-efficiency inspection methods[175,176,177]. Especially in high mountains, swamps, and other complex terrain, as well as rain, snow, ice, earthquakes, and other disaster conditions that are difficult for personnel to reach, difficult-to-find equipment damage on a tower, and other shortcomings. With the rapid development of aviation, remote sensing, and information processing technologies, the power industry can actively carry out line construction and the operation and maintenance of new technology research. Among such technology, UAVs have the advantages of operating with high flexibility and at a low cost for line erection traction and overhead line inspection[178]. UAVs are usually controlled by flyers and collect corresponding aerial images. Researchers have used the captured data to develop many automated analysis functions, such as defect detection[179], bird's nest detection, etc. However, the existing UAV inspection system still has a single technical means, cannot synchronize line defects in real time, as well as other problems. These are mainly reflected in the following points:

(1) The degree of autonomy of the inspection flight: This needs to be improved, as the inspection efficiency is low. At present, a mainstream inspection flight robot basically uses a combination of human and machine inspection, the need for the manual operation of the UAV for inspection target photography, which involves copying or first manually operating the UAV for photo point location collection, and then re-flying inspection. Photo copying requires manual participation, a low degree of autonomy, and low inspection efficiency;

(2) Flight control stability issues: An inspection flight robot in response to the complex inspection environment, has difficulty in achieving high precision and stable hovering, which brings a serious impact on accurate data collection, so flight control stability has been a difficult point for industry applications;

(3) Drone battery replacement issues: An existing inspection flight robot generally lacks the functions of fast and accurate recovery and power battery replacement, which means that inspection efficiency cannot significantly improve;

(4) Inspection data fault detection: An inspection flight robot has a low accuracy for intelligent recognition and slow generation of inspection reports.

In order to solve the above problems, we have proposed innovations in autonomous flight, autonomous path planning, autonomous battery replacement, and intelligent detection and designed a new UAV inspection system, as shown in Figure 4-1. The main contributions of this chapter are summarized as follows:

(1) The ground station system that automatically generates the inspection program is designed, including fine inspection, arc-chasing inspection, and channel inspection, and the UAV can operate autonomously according to this plan to achieve the all-around inspection of high-voltage lines;

(2) The self-developed flight control and navigation system achieves high robustness and high precision flight control for the UAV, solving the problem of poor flight control stability for existing inspection robots;

(3) A mechanical device for automatic battery replacement is designed, and a mobile inspection scheme is provided to complete the transfer of equipment while the UAV performs its task, greatly improving the efficiency of inspection;

(4) Based on the YOLOX object detection model, some improvements are proposed, and the improved YOLOX is deployed on the cloud server to improve detection accuracy.

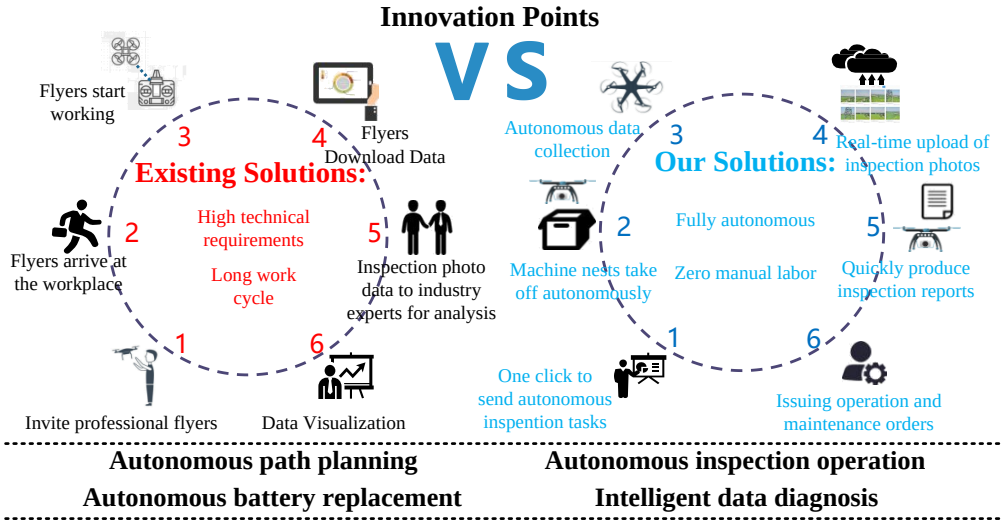


Figure 4-1: Comparison of the traditional inspection solution and our solution.

4.2 Related Work

The traditional inspection method for high-voltage lines is that inspectors inspect the lines at high a altitude, which is still used in some areas. However, this is very dangerous for personal safety because they are likely to fall from height or die by electrocution, while also working very inefficiently[180,181]. Another method is that inspectors use binoculars to check the lines, which guarantees the safety of the operators, but the inspection is also very slow[182,183]. In recent years, UASs have been playing an increasingly important role in high-voltage line inspections. Li et al.[184] proposed an unmanned intelligent line inspection system applied to the transmission grid, pointing out the construction elements, operation mechanism, and data flow diagram of the unmanned system. Calvo et al.[185] proposed a path planning scheme for UAV inspection in a high-voltage line scenario with reasonable planning for both vehicle and operator tasks, but the reliability of the system was only verified by simulation. Luque-Vega et al.[186] proposed a quadrotor helicopter-based

UAV inspection system for high-voltage lines to facilitate the qualitative inspection of high-voltage lines by power inspection departments. The UAV intelligent inspection system proposed by Li et al.[187] provided a new and efficient control and data processing method, enhanced the coordination and cooperation of UAV inspection departments, and improved the informationization and automation of UAV inspection. Guan et al.[188] proposed the concept of intelligent power line inspection by UAV with LIDAR, with a system that is able to inspect power lines with great efficiency and at a low cost, but ignores the inspection of the other components on high-voltage lines.

With the development of computer vision technology, object detection is also gradually being applied to all aspects of life, such as high-altitude vehicle detection and pedestrian detection. The mainstream object detection methods are divided into two types; one is the one-stage detection method, such as YOLO and SSD[33,35,34,36,38]. The other is the two-stage detection method, such as Faster RCNN[29]. The two-stage inspection method is highly accurate but slow, while the one-stage inspection method is fast but slightly less accurate. However, the one-stage inspection method has developed rapidly and now achieves almost the same accuracy as the two-stage inspection method. In recent years, many high-voltage line inspection projects have been combined with object detection, and many inspection functions, such as line detection[189], bird's nest detection, and insulator detection, have been developed based on various datasets. Li et al.[190] compared the performance of YOLOv3, YOLOv5s, and YOLOX_s models and proposed an optimized YOLOv5s bird's nest detection model, but the model was deployed on UAVs, which have certain real-time requirements, so the detection accuracy is not very high. Hao et al.[191] proposed a bird's nest recognition method using a combination of a single-shot detector and an HSV color space filter to further improve the accuracy of bird's nest detection. Nguyen et al.[192] proposed a method based on the combination of a single shot multibox detector and deep residual networks, capable of detecting common faults in electrical components, such as cracks in poles and cross-arms, damage on poles caused by woodpeckers, and missing top caps. However, this method is mainly used for low-voltage ordinary transmission lines and cannot be directly used for the detection of high-voltage line faults. Yang et al.[193] combined deep learning and migration learning approaches to propose a new aerial image defect recognition algorithm that can better detect insulators in complex environments.

4.3 Structure of The System and Methods

In this section, firstly, the overall structure of the system is described. Next, the generation of the scheme in the ground station system is described (path planning). Then a strong robust flight control algorithm is designed to make the UAV fly stably even during high-altitude operation. Next, a mobile inspection scheme is introduced to improve the inspection efficiency. Finally, based on the basic framework of YOLOX[194], some optimization schemes are designed to improve the model's detection accuracy.

4.3.1 Structure of the System

The structure of the system in this chapter is shown in Figure 4-2. Firstly, the operator needs to request the basic data of the high-voltage towers from the ground

station and generate inspection tasks to send to the UAV. After the drone's self-inspection is completed, upon receiving the start command, it begins to perform the operation task and inspect the electric tower. The drone inspection process uploads the photos of the inspection target to the cloud server in real time. After receiving the photos of the inspection, the cloud server uses a combination of manual and deep learning to detect the photos from the inspection. Manual detection is mainly for when they are some defects in the line, while intelligent detection is mainly for the detection of bird's nests in high-voltage lines, and the inspection report is generated after the detection is completed. After viewing the report, the staff can arrange maintenance personnel to carry out maintenance. After the drone completes its task, it returns to the intelligent machine nest, which will replace the drone's battery autonomously to improve the inspection efficiency and prepare for the next inspection task.

4.3.2 Path Planning

Inspection drones operate autonomously according to the mission plan planned by the ground station. Using the inspection equipment carried by the UAV, the inspection demand points are photographed, and the high-voltage line inspection is completed efficiently; and its inspection demand is shown in Table 4-1. According to this demand, this chapter designs three path planning schemes for fine inspection, arc-chasing inspection, and channel inspection.

Fine Inspection

According to the inspection requirements of No. 1–10 in Table 4-1, a fine inspection scheme is designed, as shown in Figure 4-3. Each task point of the path planning is calculated by the base data of the electric tower in the database. The base data of the tower include latitude, longitude, height, directional angle, and the category of the tower. As shown in Figure 4-3, the direction perpendicular to the azimuth of the tower is the azimuth of the task point location. Taking mission points 2 and 8 as examples, their latitude and longitude can be obtained from Equations (4-1) and (4-2). X_0 and Y_0 are the latitude and longitude of the center point of the current tower. D is the distance of the task point from the center point, determined by the length of the cross-arms of the tower and the safety distance, and the plus and minus signs indicate both sides of the tower. θ is the azimuth of the current tower. T_X and T_Y are the conversion factors between actual distance and latitude and longitude at the current latitude and longitude. The mission point altitude can be obtained from Equation (4-3). The above method can obtain the 3D information for task point locations 2 and 8. Task points 3, 4, 5, 9, 10, and 11 can be based on the height of task points 2, and 8, minus the height of the cross-arms. Mission points 6 and 7 are determined by adding a certain safety distance (8 m) to mission points 2 and 8, to ensure that the UAV safely crosses the high-voltage lines.

$$X = X_0 + ((\pm D) * \frac{cos(\theta - 90)}{T_X} \tag{4-1}$$

$$Y = Y_0 + ((\pm D) * \frac{sin(\theta - 90)}{T_Y} \tag{4-2}$$

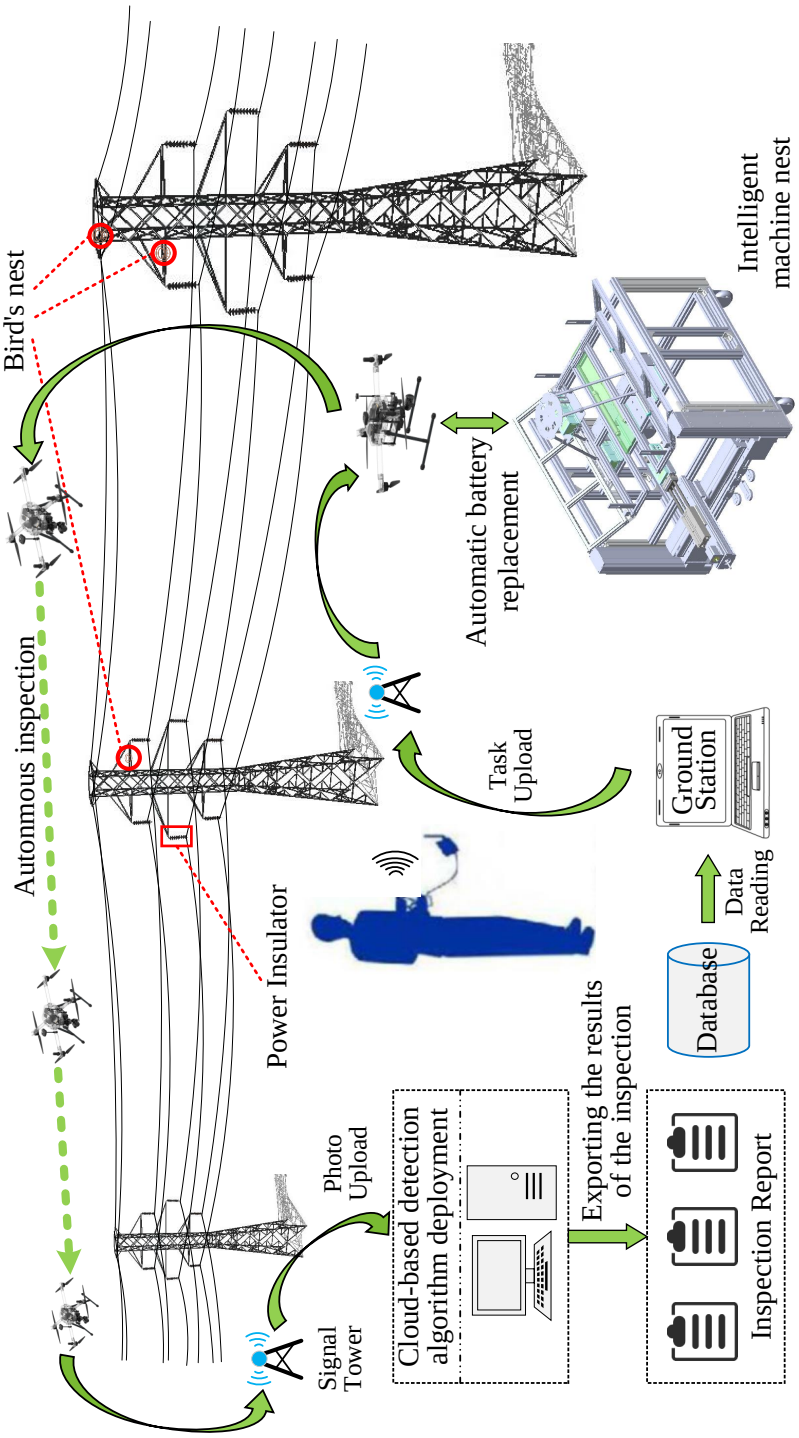


Figure 4-2: The structure of the system.

Table 4-1: High-voltage power transmission line inspection demands.

No.	Point of Demand	Inspection Contents
1	High-voltage line towers	Detection of whether the tower is deformed or tilted, the presence of a bird's nest, etc.
2	Tower bases	Detection of the ground conditions near the tower base
3	Cross-arms	Detection of whether the cross-arms are tilted and other abnormalities
4	Insulators	Detection of an insulator skirt and grading ring damage
5	Bolts	Detection of whether the installed bolts and nuts have popped out or fallen off, etc.
6	Lightning rod and grounding device	Detection of whether the discharge gap between them has changed significantly
7	Anti-vibration hammer	Detection of the fracture of the anti-vibration hammer connection
8	Lead wire pegging point fixtures	Testing of small size fixtures such as wire pendant locking pins
9	Ground wire	Detection of ground wire for loose strands and other defects
10	Ground wire pegging point fixtures	Detection of ground peg locking pin and other objects
11	Power transmission lines	Detection of whether the transmission line is broken, damaged by foreign objects, etc.
12	Channel	Checking of over-height trees and illegal buildings in the passage

$$H = H_0 \pm h \tag{4-3}$$

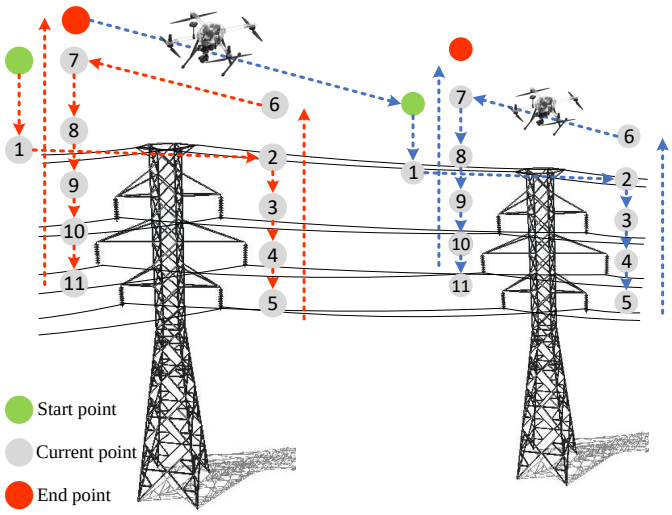


Figure 4-3: The process of fine inspection.

According to the fine inspection scheme shown in Figure 4-3, the specific inspection strategy is designed as follows: Starting the inspection task from the ground, when the inspection drone reaches the starting point position, it starts to descend in height to task point 1, i.e., after the same height as the ground line, it performs the task of taking pictures at that point. Then, it proceeds to mission point 2 at the ground wire and takes a picture of the ground wire. After the ground line photo operation is completed, then the inspection drone's height is lowered to reach task point 3 at the upper phase position of the tower, task point 4 at the middle phase position, and task point 5 at the lower phase position, to complete the photo task corresponding to each corresponding point. When the tasks of the single-side tower are finished, the inspection drone is raised to cross-tower task point 6 and reaches task point 7 on the opposite side of the tower by moving laterally. There is no photo task at these two points, so the role is to allow the inspection drone to traverse towers at a safe height. On the other side of the tower, the inspection drone lowers its altitude to reach task points 8 to 11 and complete the photo task. When all the tasks of the first tower are performed, the inspection drone is raised to the termination point, and then it flies to the starting point of the next tower. The inspection drone continues to perform the above inspection actions according to the task data until it reaches the end of the mission; at this point, the fine inspection task is completed.

Arc-Chasing Inspection and Channel Inspection

According to requirement 11 in Table 4-1, the arc-chasing inspection scheme is designed as shown in Figure 4-4 (a), and the 3D information of the task points can be obtained as described in Section 4.3.2. By using inspection drones to perform arc-chasing inspection tasks, operators can check whether the transmission lines are broken, damaged by by foreign objects, etc. According to requirement 12 in Table 4-1, the channel inspection scheme design is as shown in Figure 4-4 (b). Through the

channel inspection task, operators can inspect the high-voltage line channel, which affects tower and line safety.

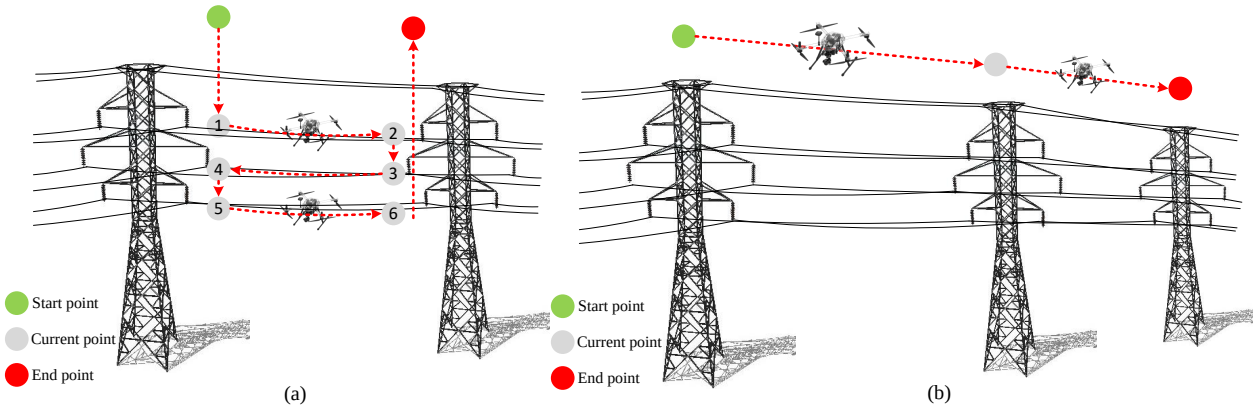


Figure 4-4: (a) The process of arc-chasing inspection; (b) The process of channel inspection.

The specific strategy for the execution of the arc-chasing inspection task is as follows: The inspection task starts from the starting point, and since there is no demand for photography at the starting point, the inspection drone descends in altitude to task point 1. Once the drone arrives at task point 1, the flight control system adjusts the camera pitch angle to take pictures, while the inspection drone flies at a certain speed to the upper phase conductor of the second tower, i.e., task point 2. When mission point 2 is reached, the on-board camera is suspended at this point because there is no photo task at this mission point. The height of the inspection drone is lowered to reach the mid-phase conductor, i.e., task point 3. Then, the on-board camera equipment is turned on again and performs the same action as above, to complete the inspection operation of the middle-phase transmission lines at task points 3 and 4 and the lower-phase transmission lines at task points 5 and 6. When reaching task point 6, the last one, the inspection drone is raised to the termination point, so that the arc-chasing inspection task is completed.

The specific strategy for the execution of the channel inspection task is as follows: The inspection task is executed from the starting point, and the starting task point is located at a fixed height directly above the first pole tower; when the inspection drone reaches the starting point, the flight control system controls the on-board camera, which starts working and takes pictures of the channel below at regular intervals. When the inspection drone reaches the second point, it continues to fly to the subsequent task points until it reaches the last termination point; then, the on-board camera stops working, the inspection drone starts to return, and the channel inspection task is completed.

4.3.3 Sliding Mode Control Algorithm

The whole UAV control system adopts the structure of position control, speed control, attitude control, and bottom stabilization control, as shown in Figure 4-5. With this approach, complex control problems can be decomposed, thus facilitating the design and implementation of the overall controller. The control objects of the position controller include the velocity controller, the attitude angle controller,

the attitude angle rate controller, and the robot's power system. When the UAV receives the latitude and longitude of the target point, sent by the ground station as the control input, it can perform position control by combining the real-time latitude and longitude information during the inspection, thus calculating the target value for speed control. Velocity control refers to the process of calculating the attitude angle target value in the UAV body coordinate system by the velocity error in the N and E directions. Since both the attitude angle controller and the attitude angle rate controller operate at a high frequency, the main characteristics of this data source are the low amount of error and stable acquisition in all environments. Therefore, the performance of the attitude angle controller as well as the attitude angle rate controller is usually relatively stable, and the performance of the speed controller directly determines the stability and accuracy of the flight process of the inspection robot.

Regarding the choice of control algorithm, the sliding mode control has strong robustness and can tolerate external disturbances well, so we chose the sliding mode control algorithm to design the speed controller of the UAV.

$$\dot{u} = -\frac{1}{m}[(sin\theta cos\varphi)\sum_{n}^{i=1} C_T\Omega_i^2 - \rho S C_r u^2] \tag{4-4}$$

Equation (4-4) is satisfied between the multi-rotor UAV motor speed Ω and the velocity $\dot{u}$, where m is the weight of the multi-rotor UAV; θ and φ denotes the pitch and roll angles of the UAV, respectively; n denotes the specific number of rotors; C_T is the lift coefficient; ρ is the air density; S denotes the windward area of the UAV in flight; and C_r is the air drag constant.

$$\begin{cases} \dot{x}_1 = a_1 x_1 + a_2 u \\ \dot{x}_2 = -g x_1 + a_3 x_2^2 \end{cases} \tag{4-5}$$

Ignoring the coupling between the axes during the motion and considering only the motion in a small angular range, the velocity model can be assumed as Equation (4-5), where $\dot{x}_1$ is the dynamic acceleration, $\dot{x}_2$ represents the real-time velocity, and a_1, a_2, and a_3 are the model parameters, which can be obtained by debugging. Since the measurement result of the speed sensor is usually accompanied by a measurement delay, we add the delayed speed to the system as an extended state, and then Equation (4-5) can be expressed again as Equation (4-6), where d is the delay factor.

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} a_1 & 0 & 0 \\ -g & 0 & 0 \\ 0 & \frac{2}{d} & -\frac{2}{d} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} a_2 \\ 0 \\ 0 \end{bmatrix} u \\ &= AX + Bu \end{aligned} \tag{4-6}$$

Based on this model, the reference model for designing the speed control is shown in Equation (4-7), where r is the original velocity target information and the output matrix in the reference model is consistent with the real model, i.e., $C_m = C$, while the input matrix $B_m = B\left(-C_m A_m^{-1} B\right)^{-1}$, the specific calculation procedure of which is described in Ref.[195]. A_m can be obtained by debugging.

$$\begin{cases} \dot{X}_m = A_m X_m + B_m r \\ Y_m = C_m X_m \end{cases} \tag{4-7}$$

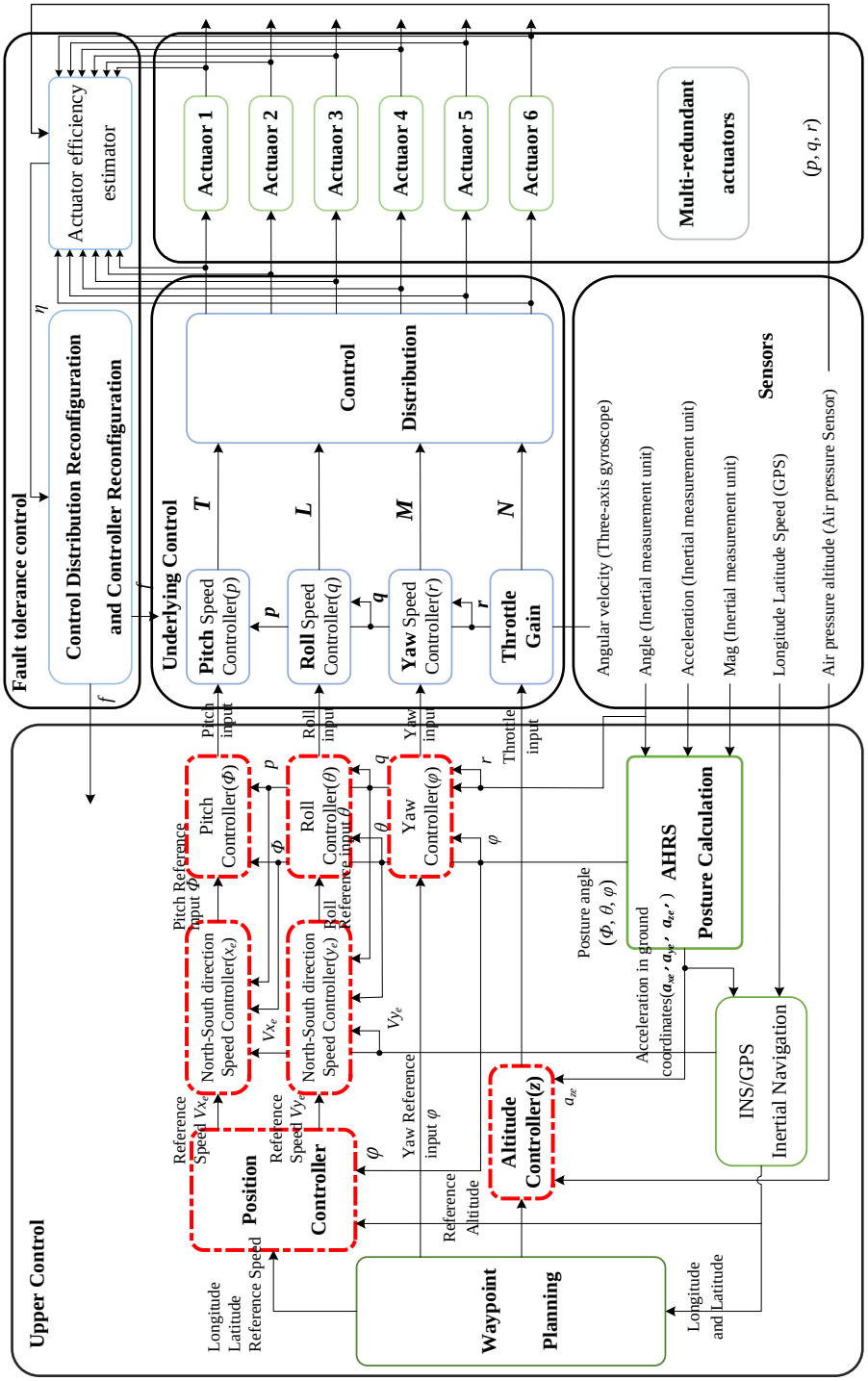


Figure 4-5: The structure of the control.

By designing the reference model, the original velocity information can be transformed into the target value corresponding to each state, and the error between the actual state and the target state is noted as Equation (4-8), where $u_s = -(K_1X - K_2r - u)$ and ε is the integral of the velocity error. The design switching function is $\sigma = Se_s$, and the derivative is shown in Equation (4-9).

$$\dot{e}_s = \begin{bmatrix} \dot{e} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A_m & 0 \\ C_m & 0 \end{bmatrix} \begin{bmatrix} e \\ \varepsilon \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_s \tag{4-8}$$

$$\begin{aligned} \dot{\sigma} &= S\dot{e}_s \\ &= SA_{mn}e_s - SB_m\left(K_1X + K_2r - u\right) \end{aligned} \tag{4-9}$$

After each state converges and remains in the sliding mode plane, the switching function and its derivative are zero, and then its equivalent control can be expressed as Equation (4-10).

$$u_{eq} = -\left(SB_m\right)^{-1} SA_m e_s + K_1X + K_2r \tag{4-10}$$

$$u_{nl} = K_n f(\sigma) \tag{4-11}$$

$$u = u_{eq} + u_{nl} \tag{4-12}$$

To avoid chattering, we chose to use the smooth function $f(\sigma) = \sigma/(|\sigma| + \delta)$ instead of the symbolic function in the traditional control, as shown in Equation (4-11). The total output of the final sliding mode controller is shown in Equation (4-12).

4.3.4 Intelligent Machine Nest

To improve the efficiency and inspection time, this chapter designs a mobile inspection scheme, as shown in Figure 4-6. When the inspection drone starts to perform the task, the operator drives the vehicle to the ready landing position in advance and waits to replace the battery after the inspection drone work is completed. Transferring sites and putting away the equipment are completed during the inspection time period. The intelligent machine nest is able to charge the drone's battery, which guarantees that the drone can carry out long inspection missions.

The whole structure of the intelligent machine nest is shown in the lower right corner of Figure 4-6. The upper surface is the apron, its structure with beveled edges on the left and right can make up for the accuracy error of the inspection drone landing on the apron, and four sensor brackets are installed on the apron to detect whether there is an inspection drone on the apron. Four cylinders are installed on the lower surface of the apron, which is the power equipment for the homing and locking device of the inspection drone. Six battery compartments are installed underneath the cylinders, which are fixed on two crossbeams. The space underneath the battery compartments is reserved for the optional installation of charging stewards, chargers, and air compressors. The robotic arm designed in this chapter has three degrees of freedom, which are the Z-axis robotic arm providing up and down degrees of freedom, the X-axis robotic arm providing left and right degrees of freedom, and the Y-axis robotic arm providing front and rear degrees

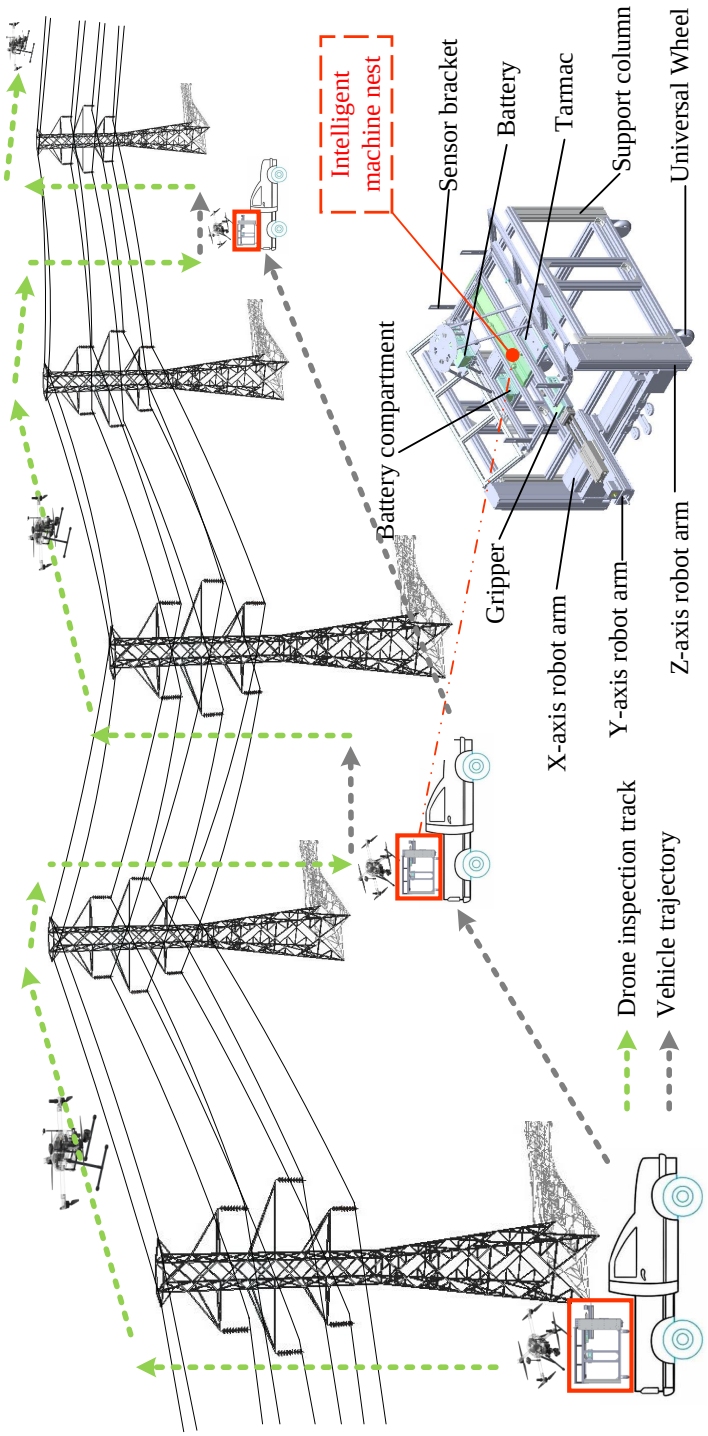


Figure 4-6: Intelligent machine nest. Green is the drone inspection route and gray is the vehicle route.

of freedom; the battery gripper is installed on the Y-axis robotic arm. Universal wheels are installed on the four corners of the bottom of the intelligent machine nest to form a mobile and fixed device of the intelligent machine nest.

4.3.5 YOLOX

YOLOX uses YOLOv3 as the baseline, with Darknet53 backbone architecture and spatial pyramid pooling (SPP) layer. The main contribution of YOLOX is the introduction of the "Decoupled Head", "Data Augmentation", "Anchor Free", and "SimOTA Sample Matching" methods. An anchor-free end-to-end object detection framework is built and achieves top-level detection.

Decoupled Head is a standard configuration in object detection one-stage networks, such as RetinaNet[196], FCOS[55], etc. The final bounding box in YOLOv3 is implemented together with the confidence in a 1×1 convolution, while in YOLOX the confidence and regression boxes are implemented separately by decoupling the header and being combined into one at the prediction time. Decoupling the detection head increases the complexity of the operation; in order to achieve a balance between speed and performance, the experiments first used one 1×1 convolution to reduce the dimensionality and then used two 3×3 convolutions in each of the classification and regression branches, which ultimately allowed the model to increase the parameters only a little and brought a 1.1 percentage point improvement in AP on the COCO dataset. YOLOX uses the Mosaic and MixUp data enhancements, which add 2.4 percentage points to YOLOv3. It should be noted that these two data enhancements were turned off for the last 15 epochs of training; before that, Mosaic and Mixup data enhancements were turned on. It was found that ImageNet pre-training would be meaningless due to a stronger data enhancement approach, so all models were trained from scratch. YOLOX uses the Anchor Free method to reduce the model parameters. From the original three groups of anchors predicted by one feature map to one group, the coordinate value of the upper left corner of the grid and the height and width of the predicted box are predicted directly. The main role of SimOTA is to assign a ground truth box to each positive sample in the output prediction box of the network and let the positive sample fit that ground truth box. This replaces the previous anchor scheme to fit the anchor, thus achieving anchor free. SimOTA enables YOLOX to improve 2.3 percentage points on the COCO dataset.

4.3.6 Improved YOLOX_m

In order to improve the accuracy of the model, the following improvements are made in this chapter based on the YOLOX_m network structure. Firstly, coordinate attention (CA)[197] is introduced after the output feature map of backbone, which embeds the location information of the feature map into the channel attention. Then, the binary cross entropy (BCE) Loss in the confidence loss is changed to the varifocal loss (VFL)[198], to solve the problem of the low confidence of the box where the location prediction is very accurate, i.e., the problem of unbalanced positive and negative samples. Finally, the SCYLLA-IoU (SIoU)[199] loss function is introduced to improve the capability of the bounding box regression. We also tried to add adaptively spatial feature fusion (ASFF)[200] after the output feature map of the neck, but the accuracy improvement on the validation set was very small and added

a larger computational effort, so the trick was not increased. The structure of the improved YOLOX is shown in Figure 4-7.

Coordinate Attention

In the field of object detection, attention mechanism is a very common trick. The more commonly applied attention mechanisms are squeeze-and-excitation networks (SENet)[166], the convolutional block attention module (CBAM)[201], efficient channel attention (ECA)[202], and coordinate attention (CA)[197]. The main idea of SENet is to refine the values on the long-width dimension into a single value and then multiply it by the original value on top of the long-width, thus enhancing the useful information and suppressing the less-useful information. CBAM can be considered as an enhanced version of SENet, where the main idea is to perform attentional operations on features in space and on channels. ECA builds on the SENeT module by changing the use of the fully connected layers in SENet to learn the channel attention information for the 1×1 convolutional learning of channel attention information. This avoids channel dimensionality reduction when learning channel attention information, while reducing the number of parameters.

Coordinate attention is mainly divided into coordinate information and coordinate attention generation. The specific structure is shown in Figure 4-8. For the input feature map x, the channels are first encoded along the horizontal and vertical coordinate directions using pooling kernels of dimensions $(H, 1)$ and $(1, W)$, respectively. Therefore, the outputs in two different directions are shown in Equations (4-13) and (4-14), respectively. The above two transformations not only return a pair of direction-aware attention graphs but also allow the attention module to capture the dependencies in one direction, while preserving the position information in the other direction, which allows the network to localize the target more accurately.

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h, j) \tag{4-13}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le i \le H} x_c(j, w) \tag{4-14}$$

To make full use of the above information, the two feature maps generated by the previous module are first cascaded and then a shared 1×1 convolutional transform F_1 is used, as shown in Equation (4-15). The generated $f \in \mathbb{R}^{\frac{C}{r} \times (H+W)}$ is an intermediate feature map of the spatial information in two directions, and denotes the downsampling scale.

$$f = \delta\left(F_1\left(z^h, z^w\right)\right) \tag{4-15}$$

Then, f is divided into two separate tensors, $f^h \in \mathbb{R}^{\frac{C}{r} \times H}$ and $f^w \in \mathbb{R}^{\frac{C}{r} \times W}$, along the spatial dimension. Next, the number of channels of f^h and f^w are transformed to match the number of channels of input X using two 1×1 convolutions F_h and F_w, as shown in Equations (4-16) and (4-17).

$$g^h = \sigma\left(F_h\left(f^h\right)\right) \tag{4-16}$$

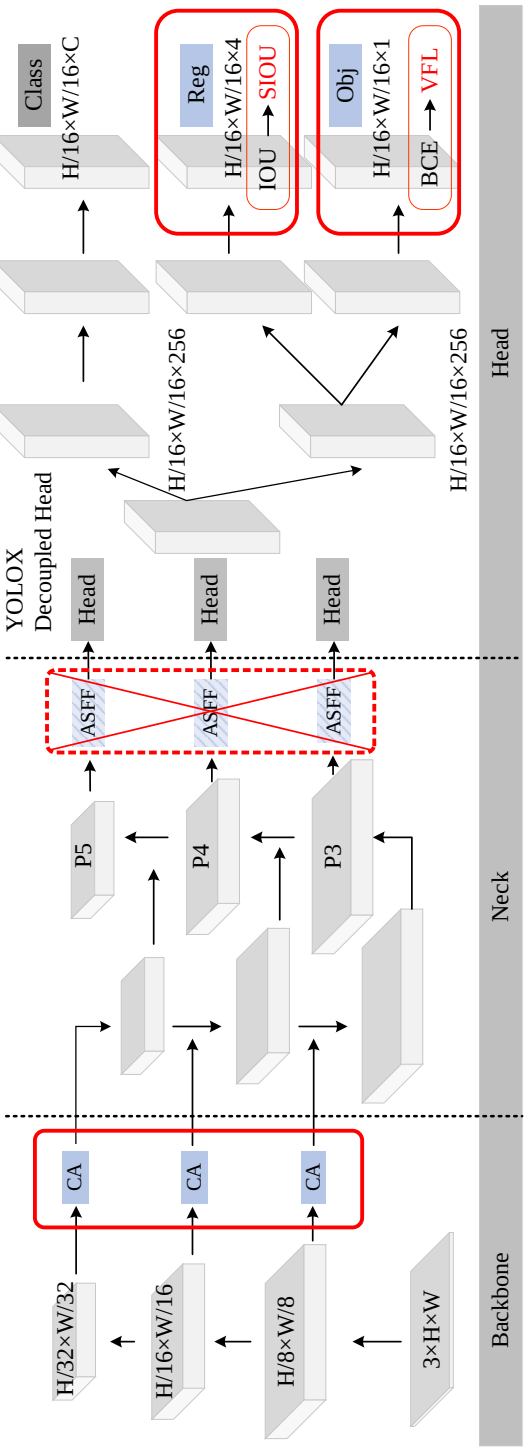


Figure 4-7: The structure of the improved YOLOX.

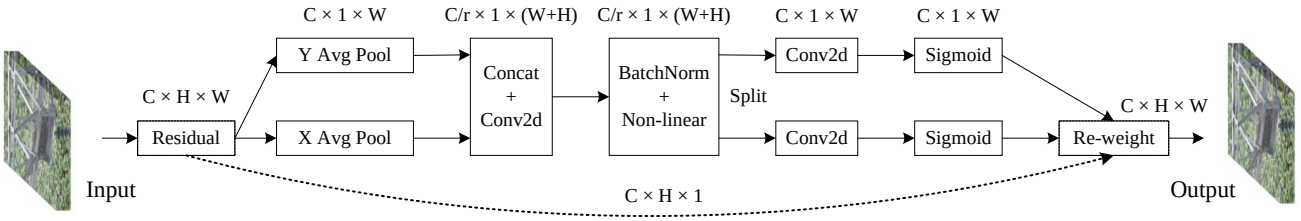


Figure 4-8: The structure of the coordinate attention.

$$g^w = \sigma\left(F_w\left(f^w\right)\right) \tag{4-17}$$

Finally, g^h and g^w are expanded as weights, and the output of the final CA module is shown in Equation (4-18). It is important to consider that when the model introduces the attention mechanism, the number of input and output channels on the feature map should be consistent with the original network.

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{4-18}$$

Varifocal Loss

The confidence loss in YOLOX is the binary crossentropy (BCE) loss, and the BCE is defined as in Equation (4-19), where y_i is the binary label value 0 or 1, and $p(y_i)$ is the probability of belonging to the y_i label value. When the label value $y_i = 1$, $BCELoss = -logp(y_i)$, the label value $y_i = 0$, and $BCELoss = -logp(1 - y_i)$. It can be seen that the loss is small when the predicted value is close to the labeled value and large when the predicted value is far from the labeled value.

$$\text{BCELoss } = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \cdot \log p\left(y_i\right) + \left(1 - y_i\right) \cdot \log\left(1 - p\left(y_i\right)\right)\right] \tag{4-19}$$

However, BCE does not solve the problem of unbalanced sample classification very well, so focal loss was proposed based on BCE. Focal loss adds a moderator to reduce the weight of easy-to-classify samples based on the balanced BCE loss function, which focuses on the training of difficult samples. It is defined as Equation (4-20), where α is the weight used to balance positive and negative samples, $(1-p)^\gamma$ is the adjustment factor, and γ is the adjustable focusing parameter. The larger the value of γ is, the smaller the loss of the positive samples is, and the model's attention is directed to the hard-to-classify samples; and a large γ expands the range of samples for which a small loss is obtained. This loss function reduces the weight of the easy-to-classify samples and focuses on the hard-to-classify samples.

$$FL(p,y) = \begin{cases} -\alpha(1-p)^\gamma \log(p) & \text{if} \quad y = 1 \\ -(1-\alpha)p^\gamma \log(1-p) & \text{otherwise} \end{cases} \tag{4-20}$$

Based on this idea of weighting in focal loss, Zhang et al. used VFL to train the regression continuous IoU-aware classification score (IACS). Focal loss is treated the same for positive and negative samples, while VFL is not equivalent, and VFL is defined as shown in Equation (4-21).

$$VFL(p,q) = \begin{cases} -q(q\log(p) + (1-q)\log(1-p)) & q > 0 \\ -\alpha p^{\gamma}\log(1-p) & q = 0 \end{cases} \tag{4-21}$$

Here, p is the predicted IACS and q is the target IoU score. q is the IoU between the prediction box and the ground truth box for positive samples, and q is 0 for negative samples. VFL attenuates only the negative samples with p^{γ}, while the positive samples are weighted using q. If the positive samples have a high IoU, the loss should be larger, so that the training can focus on the samples with high quality. To balance the overall positive and negative samples, VFL also used α for weighting the negative samples.

SCYLLA-IoU

Intersection over union (IoU) loss is the most common loss function in object detection. The IoU loss defines the intersection ratio of the ground truth box and the prediction box, and the loss is 1 when there is no intersection between the prediction box and the ground truth box. However, when the prediction box is closer to the ground truth box, the loss is smaller, and when the prediction box and the ground truth box intersection and ratio are the same, the IoU loss cannot determine which prediction box is more accurate. Generalized intersection over union (GIoU)[150] loss proposes an external rectangular box and an intersecting rectangular box to better reflect the overlap between the two, which solves these two problems to some extent. However, when the prediction box is parallel to the ground truth box, GIoU loss degenerates to IoU loss. Distance-IoU (DIoU)[203] loss introduces a penalty term to directly minimize the normalized distance between the prediction frame and the center point of the ground truth box, which not only solves the nonoverlapping problem but also converges faster. Complete-IoU (CIoU)[203] loss adds a width-to-height ratio constraint over DIoU loss, which allows CIoU to have faster convergence and a further improvement in accuracy.

None of the above loss functions consider the angle, but the angle can indeed affect the regression, so the SCYLLA-IoU (SIoU) loss function was proposed by Gevorgyan et al. The SIoU loss function consists of four cost functions: angle, distance, shape, and IoU. The angle cost is shown in Figure 4-9 (a), where B is the prediction box and B_{GT} is the ground truth box. When the angle $\alpha \leq \pi/4$ from B to B_{GT} converges to α, the opposite converges to β. The maximum value is obtained at $\alpha = \pi/4$. The specific definition is shown in Equation (4-22).

$$\Lambda = \sin(2\alpha) \tag{4-22}$$

Distance cost is defined in Equations (4-23) and (4-24). Taking the horizontal direction as an example, that is, when the two boxes are nearly parallel, α tends to 0, so that the calculated angular distance between the two boxes is close to 0; at this time γ is also close to 2, and then the distance between the two boxes for the overall loss of the contribution becomes less. In addition, when α tends to 45°, the angle cost between the two boxes is calculated to be 1; at this time γ is close to 1, and the distance between the two boxes should be taken seriously and needs to account for a larger loss.

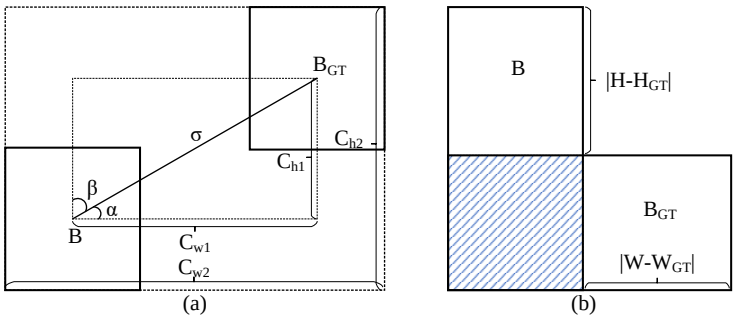


Figure 4-9: (a) Graphical explanation of SIoU loss function; (b) The definition of IoU.

$$\Delta = \sum_{t=x,y} \left(1 - e^{-\gamma p_t}\right) \tag{4-23}$$

$$p_x = \left(\frac{C_{w1}}{C_{w2}}\right)^2, p_y = \left(\frac{C_{h1}}{C_{h2}}\right)^2, \gamma = 2 - \Lambda \tag{4-24}$$

Shape cost is defined in Equations (4-25) and (4-26). Shape cost shows whether the prediction box is consistent with the ground truth box in terms of length and width, using $\theta = 4$ in the experiment. In summary, the specific definition of the SIoU loss function is given in Equation (4-27).

$$\Omega = \sum_{t=w,h} \left(1 - e^{-w_t}\right)^{\theta} \tag{4-25}$$

$$\omega_w = \left(\frac{|W - W_{GT}|}{\max\left(W, W_{GI}\right)}\right), \omega_h = \left(\frac{|H - H_{GT}|}{\max\left(H, H_{GI}\right)}\right) \tag{4-26}$$

$$SIoU = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{4-27}$$

4.4 Experiments

In this section, the dataset for the experiments, the evaluation metrics of the model, and the training conditions are presented first. Then the ablation experiments are performed on the YOLOX network model. Finally, a practicality validation test of the system is performed.

4.4.1 Dataset Establishment

The datasets in this chapter were partly obtained by autonomous UAV flights and partly collected from the Internet. Since there are fewer datasets for problems related to defects such as displacement of the grading ring and defective locking pins, the current dataset involved in this experiment is mainly about bird's nests on electric towers. A total of 2822 images of bird's nest data were collected, divided into a training set of 2430 images, a validation set of 282 images, and a test set of 110 images. In our experiments, we found that Mosaic data augmentation is not applicable to the dataset in this chapter, so we turned off Mosaic data augmentation during the model training. We also found that by adding the L1 loss function at the beginning of the training, the model performs a little better on the test set.

4.4.2 Evaluation Metrics

The evaluation metrics of the object detection model in this experiment are *Precision*, *Recall*, $mAP_{0.5}$ and $mAP_{0.5:0.95}$. *Precision* is able to detect the performance of the network model in predicting positive samples, i.e., how many of the positive samples predicted by the network model are correct positive samples; the higher the *Precision* value is, the higher the accuracy of the model detection is. *Recall* is the proportion of true positive samples predicted as positive by the network model to the total positive samples. In general, the values of *Precision* and *Recall* are mutually constrained: the higher the *Precision* is, the lower the *Recall* is, and vice versa.

$$Precision = \frac{TP}{TP + FP} \tag{4-28}$$

$$Recall = \frac{TP}{TP + FN} \tag{4-29}$$

The area under the Precision-Recall curve is called AP, and the average value of each category is mean Average Precision (mAP). AP_S is AP for small objects: area $< 32 \times 32$, AP_M is AP for medium objects: $32 \times 32 <$ area $< 96 \times 96$, AP_L is AP for large objects: area $> 96 \times 96$. $mAP_{0.5}$ is the average value of AP for each category when the value of intersection over union (IOU) is 0.5. $mAP_{0.5:0.95}$ is the average value of mAP for different IOU thresholds (IOU = 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, and 0.95). N is the total number of each category, K is the range of values of IOU, and K denotes the current threshold of IOU. $P(K)$ and $R(K)$ represent the *Precision* and *Recall* of the network model when the threshold of IOU is K, respectively.

$$AP = \sum_{k=1}^{N} P(K)\Delta R(K) \tag{4-30}$$

$$mAP = \frac{1}{C}\sum_{k=1}^{N} P(K)\Delta R(K) \tag{4-31}$$

$$\Delta R(K) = R(K) - R(K-1) \tag{4-32}$$

4.4.3 Model Training

The experimental platform for model training in this chapter is as follows: OS is Windows 11, GPU is GeForce RTX3090, CPU is Intel(R) Core(TM) i9-12900K, application development language is Python3.8, deep learning framework is Pytorchv1.11.0, and CUDA11.3. The initial parameters of the model training are as follows: the input size of the image is 768×1280, the initial learning rate is 0.01, the epoch value of warmup is 5, the value of weight decay is 0.0005, the L1 loss function is increased from the beginning of training, and the epoch of training is 200.

4.4.4 Ablation Experiments

YOLOX has six different versions of the network model YOLOX_s, YOLOX_m, YOLOX_l, YOLOX_x, YOLOX_tiny and YOLOX_nano. Among them, YOLOX_tiny and YOLOX_nano are lightweight models that require little computing power from the hardware platform and are very friendly for deployment on embedded platforms[204]. The network model in this chapter is deployed on a cloud server and is not particularly focused on speed. Therefore, we compared the performance of four other models on the dataset, and the results are shown in Table 4-2. As seen in Table 4-2, although the YOLOX_x model had a deep network and a relatively large number of computations and parameters, the accuracy was not the highest. YOLOX_m and YOLOX_l achieve almost the same accuracy, but the number of parameters and computation of YOLOX_m was only half of that of YOLOX_l, so we finally chose YOLOX_m as the baseline of the object detection model.

Attentional Mechanisms

In our experiments, we added different attention mechanisms after the feature maps outputted by backbone, and the specific results are shown in Table 4-3, which shows that the best results are obtained after adding CA. The improvement over the initial network is 0.62 percentage points in the $mAP_{0.5:0.95}$ metric and 0.36 and 0.22 percentage points over CBAM and ECA, respectively, with almost no increase in the number of parameters and computational effort. We also tried to add the attention mechanism to the feature pyramid, but the accuracy not only did not improve, but actually decreased. So, finally, only the CA module was added after the output feature map of backbone.

Confidence Loss

The confidence loss function in YOLOX is the BCE loss function, and we replaced it with the FL loss function and VFL loss function to verify the performance of the model, respectively, and the results are shown in Table 4-4. The FL loss function not only did not improve the accuracy of the network model but also made the model decrease by 0.39 percentage points. The VFL loss function improved by 0.87 percentage points on the $mAP_{0.5:0.95}$ metric. Since the loss function is only used during model training and does not change the structure of the model, it does not increase the computational effort or the number of parameters of the model.

Bounding Box Regression

IoU and GIoU loss functions are provided in YOLOX for bounding box regression, and we tried to verify the performance of DIoU, CIoU, and SIoU loss functions on our dataset; the specific results are shown in Table 4-5. As can be seen from Table 4-5, the performance of the SIoU loss function is optimal, with a 0.73 percentage point improvement over the IoU loss function. However, different loss functions may perform differently on different datasets, so it depends on the variation of mAP values on the validation set.

Table 4-2: Different versions of YOLOX performance comparison.

Methods	**Size**	**Par**	**Gflops**	$\boldsymbol{mAP_{0.5}}$**(%)**	$\boldsymbol{mAP_{0.5:0.95}}$**(%)**	$\boldsymbol{AP_S}$	$\boldsymbol{AP_M}$	$\boldsymbol{AP_L}$
YOLOX_s	768 × 1280	8.94 M	64.22	97.7	70.05	/	77.6	69.6
YOLOX_m	768 × 1280	25.28 M	176.94	97.8	70.63	/	75.6	70.5
YOLOX_l	768 × 1280	54.15 M	373.61	97.9	70.65	/	74.9	70.6
YOLOX_x	768 × 1280	99.00 M	676.87	97.8	70.37	/	77.3	70.0

Table 4-3: Performance comparison of different attention mechanisms.

Methods	**Size**	**Par**	**Gflops**	$\boldsymbol{mAP_{0.5}}$(%)	$\boldsymbol{mAP_{0.5:0.95}}$(%)	$\boldsymbol{AP_S}$	$\boldsymbol{AP_M}$	$\boldsymbol{AP_L}$
YOLOX_m	768 × 1280	25.28M	176.94	97.8	70.63	/	75.6	70.5
YOLOX_m + SENet	768 × 1280	25.38M	176.96	97.8	70.30	/	75.2	70.0
YOLOX_m + CBAM	768 × 1280	25.47M	177.00	97.8	70.89	/	76.7	70.7
YOLOX_m + ECA	768 × 1280	25.28M	176.95	97.8	71.03	/	76.7	70.8
YOLOX_m + CA	768 × 1280	25.36M	177.03	97.8	71.25	/	76.8	71.0

Table 4-4: Performance comparison of different confidence loss functions.

Methods	$\boldsymbol{mAP_{0.5}}$(%)	$\boldsymbol{mAP_{0.5:0.95}}$(%)	$\boldsymbol{AP_S}$	$\boldsymbol{AP_M}$	$\boldsymbol{AP_L}$
YOLOX_m + BCE	97.8	70.63	/	75.6	70.5
YOLOX_m + FL	97.7	70.22	2.8	72.4	70.4
YOLOX_m + VFL	98.3	71.50	20.5	76.6	71.2

Table 4-5: Performance comparison of different regression loss functions.

Methods	$\boldsymbol{mAP_{0.5}}$(%)	$\boldsymbol{mAP_{0.5:0.95}}$(%)	$\boldsymbol{AP_S}$	$\boldsymbol{AP_M}$	$\boldsymbol{AP_L}$
YOLOX_m + IoU	97.8	70.63	/	75.6	70.5
YOLOX_m + GIoU	97.8	70.92	/	76.7	70.7
YOLOX_m + DIoU	97.8	71.12	/	76.7	70.8
YOLOX_m + CIoU	97.8	71.18	/	76.7	70.8
YOLOX_m + SIoU	97.9	71.36	/	76.7	71.1

The performance of the improved YOLOX_m network model is shown in Table 4-6, where row 1 is the baseline and rows 2–4 are our improved model. On the $mAP_{0.5:0.95}$ metric, the final model improves 2.22 percentage points over the original network model YOLOX_m, with almost no increase in the number of parameters and computation.

4.4.5 System Validation

In order to verify the effectiveness of the autonomous inspection system for high-voltage transmission line drones, actual flight tests are very necessary. The actual flight test was conducted with the team's self-developed UAV as the hardware platform. The experimental site was a high-voltage line in Xuzhou City, Jiangsu Province, China, and the actual flight test was conducted after approval by safety management, as shown in Figure 4-10. We not only verified the single UAV autonomous inspection operation but also carried out a test of a multiple UAVs simultaneous autonomous inspection operation.

Flight Data

The inspection drone took off from an open area around the high-voltage tower and completed its operational tasks according to the inspection plan described in Section **??**. To facilitate the viewing of the data, the latitude, longitude, and altitude of the inspection drone flight were transformed into the true distance in the (N, E, D) coordinate system, as shown in Figure 4-11. Taking Figure 4-11 (b) as an example, the inspection drone took off from (0,0,0) and conducted a single-side arc-chasing

Table 4-6: Performance of the improved YOLOX_m model.

YOLO_m	**CA**	**VFL**	**SIoU**	**Par**	**Gflops**	$\boldsymbol{mAP_{0.5:0.95}}$ (%)
✓				25.28 M	176.94	70.63
✓	✓			25.36 M	177.03	71.25 (+0.62)
✓	✓	✓		25.36 M	177.03	72.12 (+0.87)
✓	✓	✓	✓	25.36 M	177.03	72.85 (+0.73)

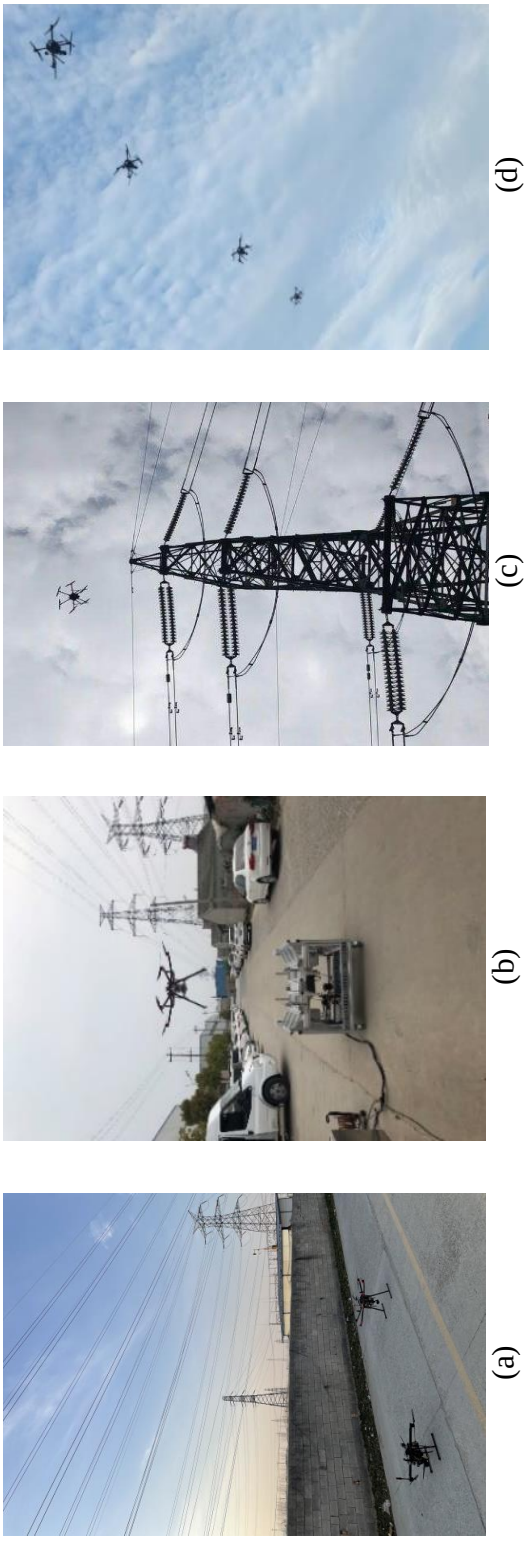


Figure 4-10: Actual flight environment. (a) Preparation stage of autonomous inspection operation of UAV; (b) Intelligent machine nest test; (c) Single UAV for autonomous inspection operation; (d) Multiple UAVs for autonomous inspection operations.

inspection of two high-voltage towers in the distance, returning to (0,0,0) after the inspection task was completed. Its real flight trajectory was consistent with the trajectory depicted in Figure 4-4 (a) in Section 4.3.2, and the flight trajectories in Figure 4-11 (a), (c) correspond to the planned trajectories in Figures 4-3 and 4-4 (b), respectively. This shows that inspection drones are able to operate precisely according to the tasks planned in the ground station.

As can be seen from Section 4.3.3, the speed control of the inspection UAV is the core of the control system, and the tracking results of the speed control can reflect the stability of the UAV flight well. The results of the speed tracking for the three inspection tasks are shown in Figure 4-12. VelN, VelE, and VelD are the values of the speed in the N direction, E direction, and vertical direction, respectively. Red dashed lines are the speed target values and blue solid lines are the actual speed values. The flight data shows that the inspection drone has good speed tracking performance and stable flight during the actual operation.

Inspection Data Collection

The schematic diagram of fine inspection data collection is shown in Figure 4-13, which shows the data pictures of arc-chasing wire, full tower, overhanging wire clip, grading ring, and insulator. These data will be uploaded to the cloud server, making it more convenient for operators to view.

A schematic diagram of the dataset collection for arc-chasing inspection and channel inspection is shown in Figure 4-14, which illustrates the specific details and surroundings of a high-voltage transmission line. The operator can check whether the high-voltage transmission line is broken and/or damaged by foreign objects according to the arc-chasing inspection data, and at the same time can observe whether there are ultra-high trees and illegal buildings in the high-voltage transmission line channel.

Results of the Bird's Nest Detection

We compared the detection results of YOLOv3, YOLOX_m, and the improved YOLOX_m model for bird's nests, from which we selected representative detection results, as shown in Figure 4-15, where the red rectangular box is the result of model detection, the interior of the yellow elliptical box is the result of model incorrect detection, and the yellow rectangular box is a zoomed-in view at the location of the yellow ellipse; the interior of the blue elliptical box is the result of model's correct detection, and the blue rectangular box is a zoomed-in view at the location of the blue ellipse. The detection sample in the first image was relatively difficult, as the YOLOv3 model did not detect the bird's nest in the image, while the YOLOX_m model detected the real bird's nest, but there was a false detection. In the second image, both the YOLOv3 model and the YOLOX_m model had false detections, while the YOLOX_m model had a relatively small range of false detections. In the third image, there were three false detections in the YOLOv3 model and one false detection in the YOLOX_m model. In the fourth image, both the YOLOv3 model and the YOLOX_m model had a false detection, but the location of the false detection were different. The improved YOLOX_m model made a correct detection for all four images. Although the confidence level of some categories is lower than

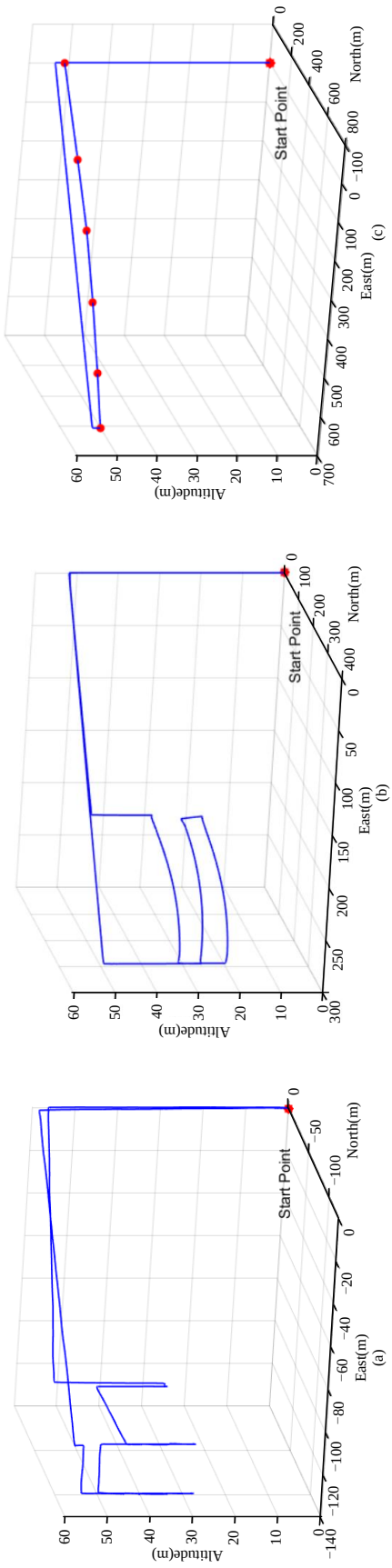


Figure 4-11: (a) The trajectory of fine inspection; (b) The trajectory of arc-chasing inspection; (c) The trajectory of channel inspection.The red points are mission points and the blue lines are the flight paths of the drones.

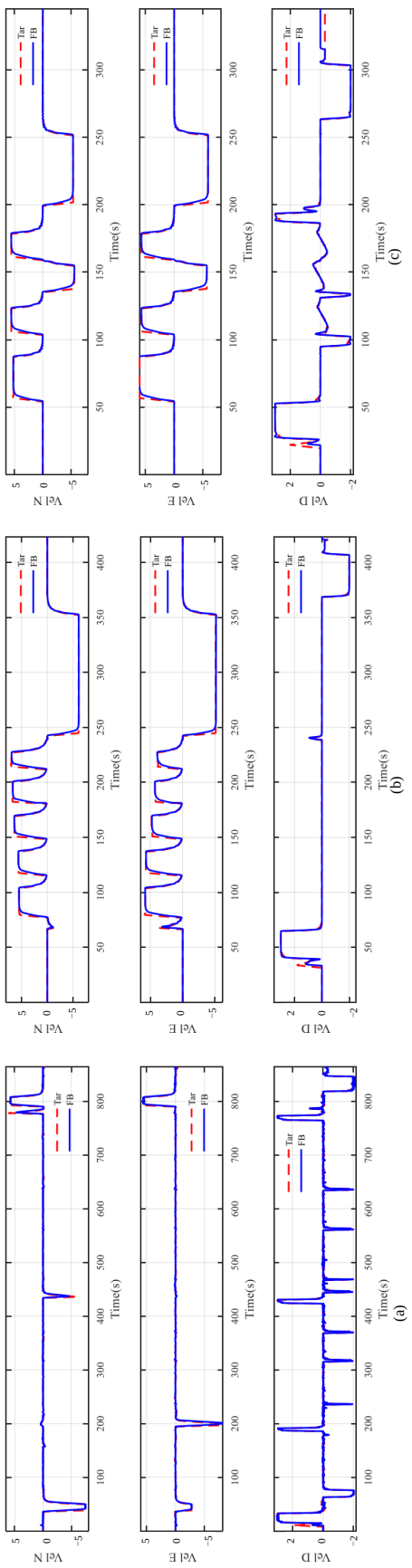


Figure 4-12: (a) Speed-tracking results for fine inspection; (b) Speed-tracking results for chasing inspection; (c) Speed-tracking results for channel inspection.

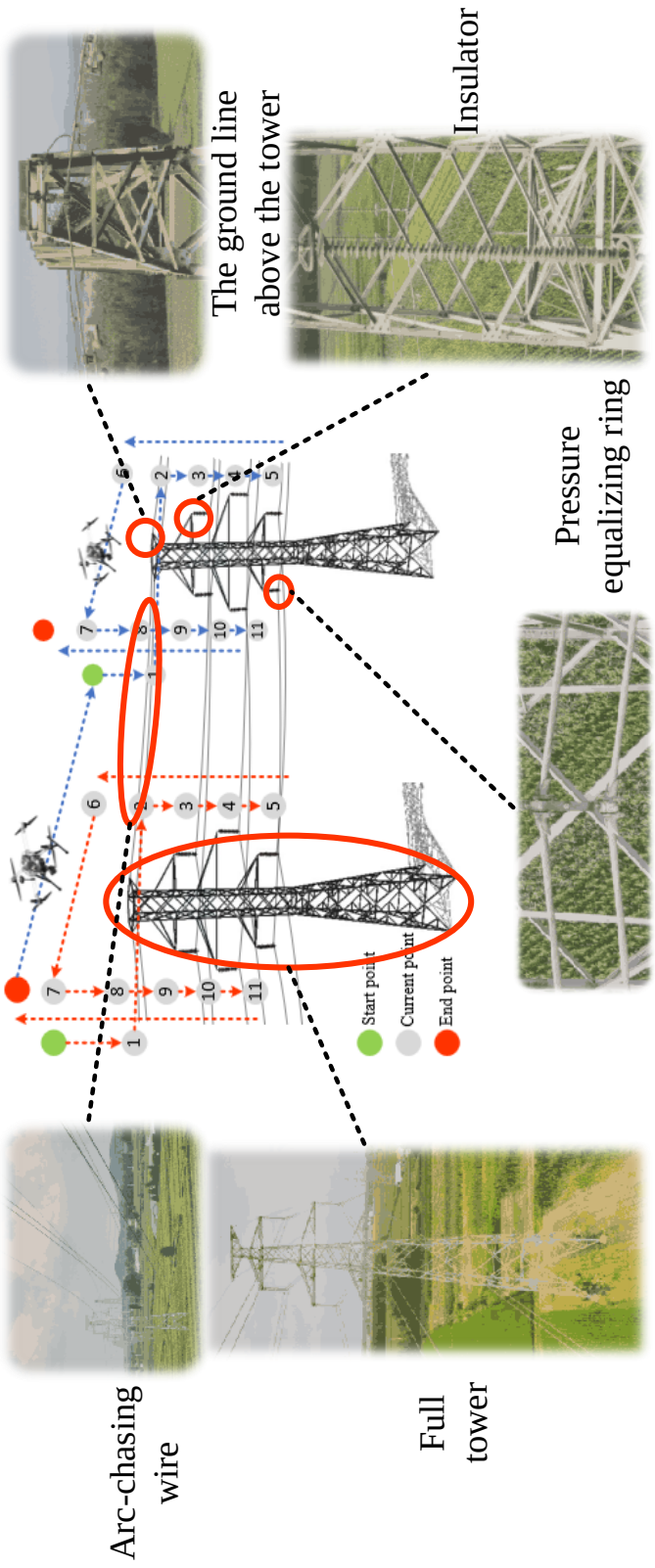


Figure 4-13: Fine inspection task data collection.

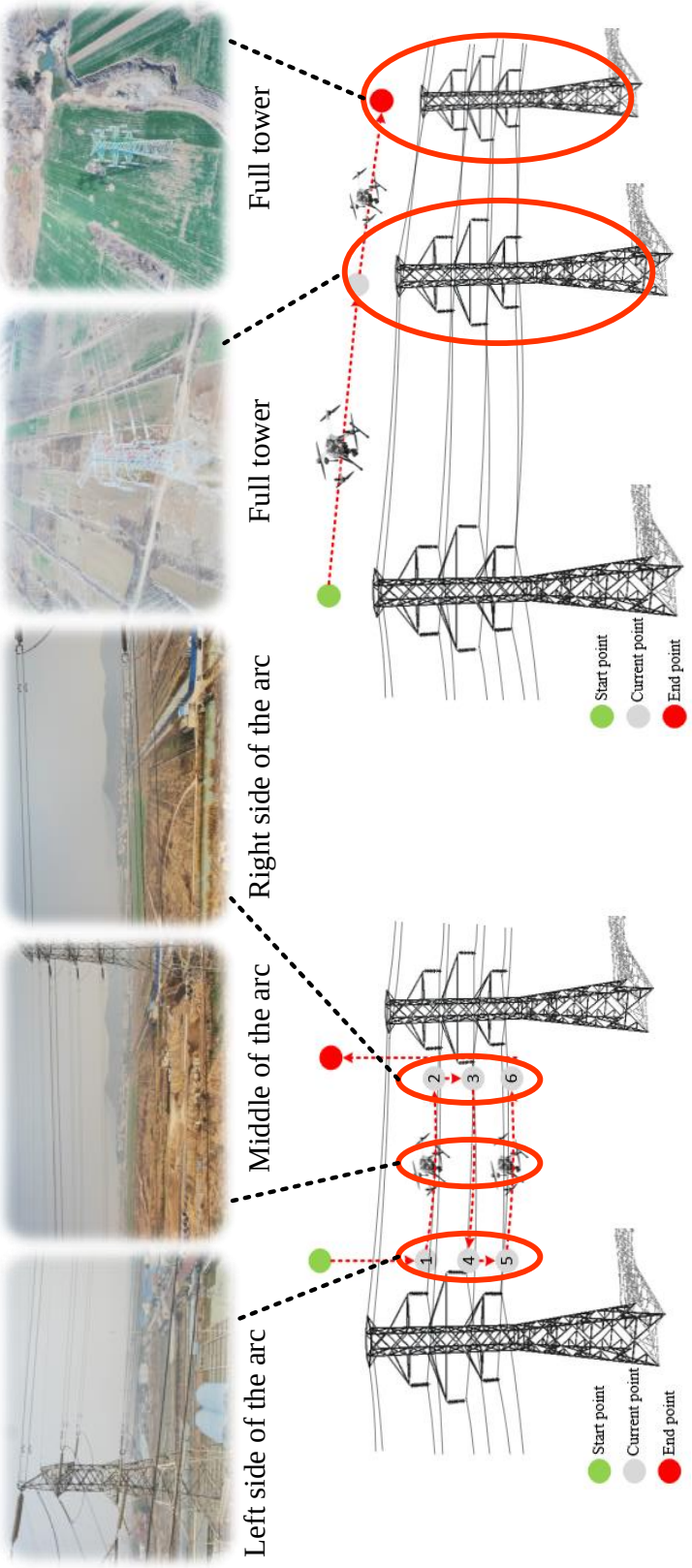


Figure 4-14: Arc-chasing inspection and channel inspection task data collection.

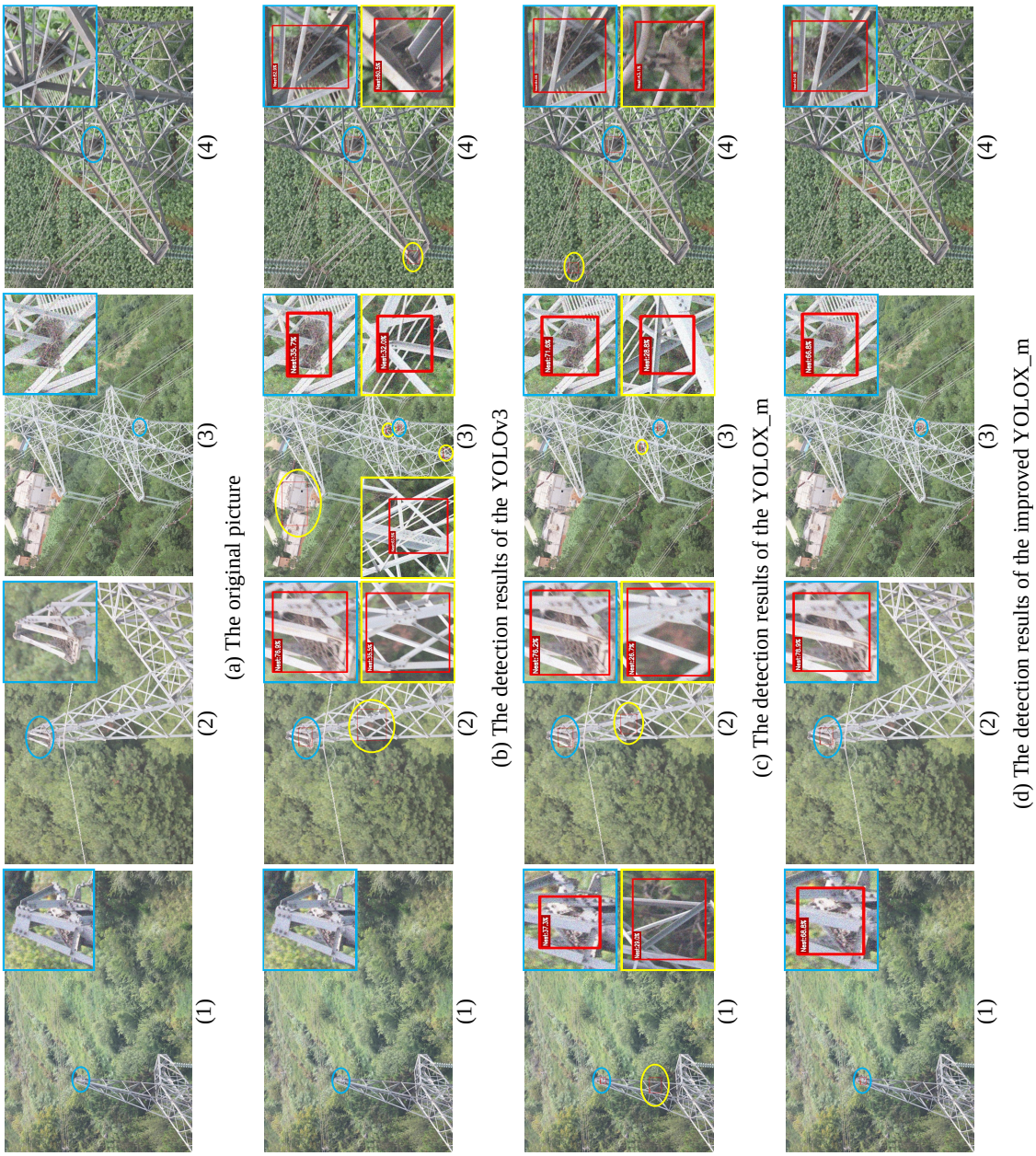


Figure 4-15: Comparison of the detection results of the different models. The red rectangular box is the detection result of the model, inside the yellow border is the incorrect detection result, and inside the blue border is the correct detection result.

the original YOLOX_m model, no wrong detections were made for any images. In conclusion, the generalization of the improved YOLOX_m model was the best.

Comparison of Inspection Efficiency

After a large number of actual flight experiments, we summarized the relevant technical indicators of this UAV inspection system and compared it with the combined human-machine inspection scheme and the traditional manual inspection scheme, as shown in Table 4-7. Compared with the human-machine combined inspection scheme, this system inspection scheme's fine inspection net time is only 5 min, saving about 10 min compared to the human-machine combined inspection scheme and saving about 40 min compared to the traditional manual inspection. The average number of pole towers inspected in a single day is 40, and the maximum number of towers inspected in a single sortie is 6, with the advantages of a duration of 42 min for a single sortie inspection and less than 3 min of intelligent machine nests for a battery replacement, plus only one staff member is needed for system monitoring. In summary, the data show that the UAS described in this chapter leads to a significant increase in inspection efficiency.

4.5 Discussion

The design of the system in this chapter is mainly considered with the the engineering application of high-voltage transmission line inspection. Compared with the existing inspection system, the advantages of the system in this chapter are as follows: (1) It designed and developed a ground station system integrating inspection path planning, task management allocation, data management, intelligent fault diagnosis, and other multi-functional functions, realizing the fully autonomous operation of the inspection process, improving the autonomy of inspection, and saving the cost investment of professional inspection operation training required by the existing inspection. (2) It independently developed a flight control system and navigation system to achieve high robustness and high precision flight control of the flying robot, solving the problem of poor stability of the existing inspection robot flight control. (3) It proposed a mobile inspection scheme, completing the autonomous battery replacement of the inspection robot on the intelligent machine nest and significantly improved the inspection efficiency. (4) It used a fusion YOLOX object detection algorithm, combined with manual detection, accomplishing the rapid generation of detailed inspection reports.

In Table 4-1, we list the many inspection requirements. At present, only the detection of the bird's nest is better, and the detection of other defects in the line cannot be detected by applying the deployed algorithm yet. This still requires manual inspection, mainly because of the limited dataset currently collected and the more complex detection of various defects, which is the drawback of this chapter. As shown in Figure 4-16, the grading ring is displaced, the insulator string is tilted, the locking pin is defective, etc. More data need to be collected and a reasonable data enhancement and neural network model needs to be applied to detect the defects.

Regarding the backbone of YOLOX, we also tried some newer backbones such as HorNet[205], EfficientFormer[206], RepVgg[207], MViT[208], etc. They have a good performance in the field of image classification, but placing them into YOLOX was

Table 4-7: Comparison of performance indicators of different inspection schemes.

Inspection Scheme Technical Index	**Our Inspection Scheme**	**Combined Human-Machine Inspection Scheme**	**Traditional Manual Inspection Scheme**
Fine inspection net inspection time	5 min	10–20 min	50 min
Average number of towers inspected per day	40	20	5
Maximum number of towers inspected in a single sortie	6	2	1
Endurance of single sortie inspection	42 min	25 min	/
Maximum inspection distance for a single sortie	5000 m	2000 m	/
Number of maximum waypoints inspected	1000	300	/
Battery replacement time	<3 min	5 min	/
Number of staff	1	3	6
Inspection report issuance time	1 day	2 days	3–4 days

Figure 4-16: (a) The grading ring is displaced; (b) The locking pin is defective; (c) The insulator string is tilted.

not very satisfactory, as there was almost no improvement for the accuracy of the model, so we kept the backbone of the original model. For the neck part of YOLOX, we also tried to add ASFF after the neck output feature map to filter the interference information and improve the amount of useful information. However, there was little improvement in the accuracy of our dataset, and a large amount of computation was added. Therefore, in the end, ASFF was not added.

4.6 Conclusion and future work

In this chapter, we designed an autonomous inspection system for high-voltage power transmission line drones, which realizes the efficient inspection of high-voltage power transmission lines. Based on the inspection demand of high-voltage power transmission lines, three path planning schemes were designed, including fine inspection, arc-chasing inspection, and channel inspection, to achieve the all-around inspection of high-voltage power transmission lines. In order to make the UAV perform stable operational tasks even at high altitude, a reference model-based sliding mode control algorithm was designed to improve flight stability. A mobile inspection solution was designed to complete the transfer of equipment during the inspection and to complete the task of automatic battery replacement at the same time, which greatly saves time and improves work efficiency. Finally, a YOLOX-based high-precision object detection algorithm was designed. Firstly, CA was added to the backbone output of the three feature maps to improve the ability of the model to extract features. Then the VFL, SIoU loss function was used to further improve the accuracy of the model. The improved YOLOX increased the $mAP_{0.5:0.95}$ metric by 2.22 percentage points for bird's nest detection. In conclusion, after a large amount of flight verification, the high-voltage transmission line UAV autonomous inspection system designed in this chapter greatly improves the inspection efficiency and reduces the cost of inspection manpower and material input. It also combines object detection technology, which makes the inspection system more intelligent.

5 UAV High-voltage Power Transmission Line Autonomous Correction System Based on Object Detection

5.1 Introduction

In recent years, drones have played an increasingly important role not only in people's daily lives but also in the rapid rise of industrial applications[209,210]. Inspection operations are areas where drones are often involved, and in many inspection operations, the drones are still manually manipulated to take pictures and store them in the SD card, and after the flight mission is over, the inspector checks the pictures. At the same time, to ensure the stable transmission of electrical energy, the high-voltage power transmission line needs to be inspected almost every year and manual inspection is time consuming and costly[211,212,213].

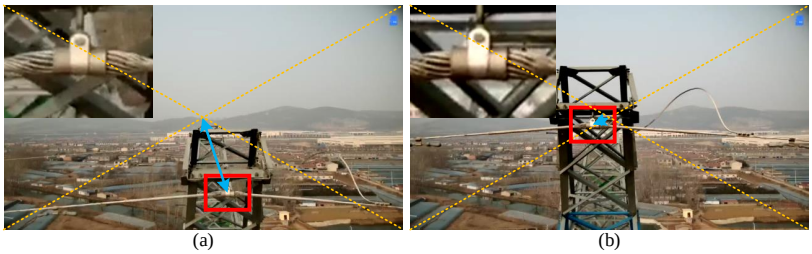


Figure 5-1: (a) The object is far from the center of the picture. (b) The object is close to the center of the picture.

In this chapter, real-time dynamic carrier phase differential technology is applied to UAVs, which rely on precise tasks issued by the central station and then fly autonomously to take pictures, saving time. However, due to the environment, electromagnetic interference, and other factors, the location point of the UAV has some error, as shown in Figure 5-1(a) (this is the figure of a dangling line clip in the high-voltage power transmission line, which protects the fiber optic cable by absorbing any auxiliary shock), because of which, the target is far from the center point of the picture. However, the results of the inspection are more inclined to the effect shown in Figure 5-1(b), making the inspection more accurate. For this reason, this chapter applies object detection to the UAV inspection task to determine the location of the object, keep the target in the center of the picture as much as possible, and then upload the high-definition picture to the server for real-time viewing by the inspectors. The main contributions of this chapter are as follows:

(1) During the autonomous operation of the drone, the problem of the target not being in the center of the picture when the drone is taking pictures at a high altitude is solved and the efficiency and accuracy of the inspection is improved;

(2) Some applied optimization schemes based on the YOLOX model are proposed, and the optimized model is applied to the UAV, which enables the UAV to obtain real-time target information;

(3) A complete correction system is designed, including a control module, a path planning module, error conversion, and some strategies.

5.2 Related Work

Traditional object detection algorithms are adapted to situations with obvious features and simple backgrounds, while in practical applications, the backgrounds are complex and variable, as are the targets to be detected, making it difficult to detect targets by means of general abstract features. Deep learning can extract rich features of the same target and complete target detection. Sometimes, the abstract features of the target cannot be summarized in various complex and variable situations. So, one has to revert to the second-best method and use the huge and rich data to complete the training of the model through deep learning, which makes the algorithm more robust, more generalizable, and easier to apply to practical scenarios. Although the method proposed by Ma *et al.*[214] shows good results for insulator detection, it has low generalizability and is not applicable to the research in this paper. Nowadays, deep-learning-based object detection can be roughly divided into three categories: (1) One-stage detection methods, such as YOLO and SSD[33,35,34,36,38]. (2) Two-stage detection methods, of which the most representative is Faster RCNN[29]. The inference speed of the one-stage detection method is relatively high, unlike the two-stage detection method, which has high localization and target recognition accuracy and relatively low inference speed. (3) The anchor-free object detection algorithm, which is divided into two main types: the Dense Prediction type represented by DenseBox[215], which intensively predicts the relative positions of boxes, and the Keypoint-based Detection type, represented by CornerNet[40], which focuses on detecting target key points. There are many areas of daily life in which object detection is required, such as overhead vehicle detection, helmet detection, and animal detection[216,217,128].

Object detection algorithms are also becoming more widespread in power inspections. For example, the Improved YOLOv3 Network proposed by Liu *et al.* has not been validated for practical deployment[218], although the model has superior performance and runs faster on GPU. The models proposed by Li *et al.*[219] and Rahman *et al.*[132] did not undergo actual flight tests, although they compared the performance of the YOLO series models on their own datasets. The improved Faster R-CNN proposed by Liu *et al.*[220] and the network model proposed by Miao *et al.*[221] although high in accuracy, have poor real-time performance and are not applicable to the real-time detection of UAVs. The model proposed by Wang *et al.*[222] is for large resolution 3968×2976 images, which are impractical to process in real time on UAVs with limited hardware resources. As mentioned above, although there are many object detection models with high accuracy, the real-time performance is relatively poor. Moreover, many researchers do not use specific data from electrical towers for path planning, so operators are required to fly manually to collect data. However, In this chapter, in addition to sensor data acquired during autonomous flight, a database of the position and attitude of electrical towers in the power grid is used as prior knowledge to achieve rational path planning and autonomous flight operations.

There is also a lot of research on UAS for electrical tower inspection. Calvo *et al.*[185] proposed a complex mission planning scheme for UAS that makes UAS more automated. The UAV inspection system proposed by Li *et al.*[184] incorporates image processing technology, which makes the inspection more intelligent. The power line inspection system proposed by Hui *et al.*[223] basically detecs and tracks targets, but the real-time performance is poor. The power line inspection system proposed by

F. Luque-Vega *et al.*[186] uses a TIR camera, which makes it more sensitive to the devices and components in the line. However, all the above systems ignore one problem: the UAV will have some accuracy error at the mission point, because of which, the target in the picture will deviate from the center, making the acquired data inaccurate.

To address the above problems, in cooperation with electric power companies, we have developed a UAV high-voltage power transmission line autonomous correction and inspection system. In this chapter, we have designed a lightweight object detection model based on YOLOX[194] to achieve real-time detection on UAS and also designed a error correction algorithm to ensure that, as far as possible, the object remains in the center of the picture. This method not only enables intelligent inspection to be more accurate and efficient but also collects higher-quality data.

5.3 Structure of The System and Methods

In this section, we first introduce the structure of the system, explain the framework of YOLOX, and make some application-specific improvements to YOLOX. We then present the algorithm of pixel error to actual error. We, finally, introduce the UAV autonomous correction inspection system.

5.3.1 Structure of The System

Figure 5-2 presents the structure of the system. First, we collect the corresponding dataset using the team's self-developed UAV. Then we train the object detection model for this experiment on a desktop computer and deploy the trained model to the Xavier NX. The system works as follows: It transfers the images captured by the camera in real-time to Xavier NX for processing. Xavier NX calculates the actual distance error based on the pixel error and transmits this error to the flight control system. The flight control system calculates the position points to be adjusted based on its own latitude, longitude, altitude, heading angle, and actual distance error. Finally, the system uses position control and altitude control to fly to the location point so that the target is as centered as possible in the picture.

The underlying system layer mainly involves camera drivers, storage drivers, hardware drivers, and drivers for various sensors. The framework layer mainly consists of the Robot Operating System (ROS), the Pytorch deep learning framework, and computer open source libraries such as OpenCV. The application layer mainly includes the flight control node, the fixed flight node, and the MAVROS communication node, where the connection between the application layer and the framework layer is established through the MAVROS communication node.

5.3.2 Improved YOLOX_tiny

In this chapter, to balance the accuracy and speed of the model, we improve the model on the basis of YOLOX_tiny: (1) To improve the generalizability of the model, we adopt reasonable data enhancement methods. (2) To reduce the amount of computation and speed up the inference of the model, the backbone of YOLOX_tiny is changed to MobileNetv3[224]. (3) To reduce the number of parameters and computation of the model under the premise of guaranteeing accuracy, we introduce the Ghost module and depthwise convolution[225]. (4) To operate with focus over a larger

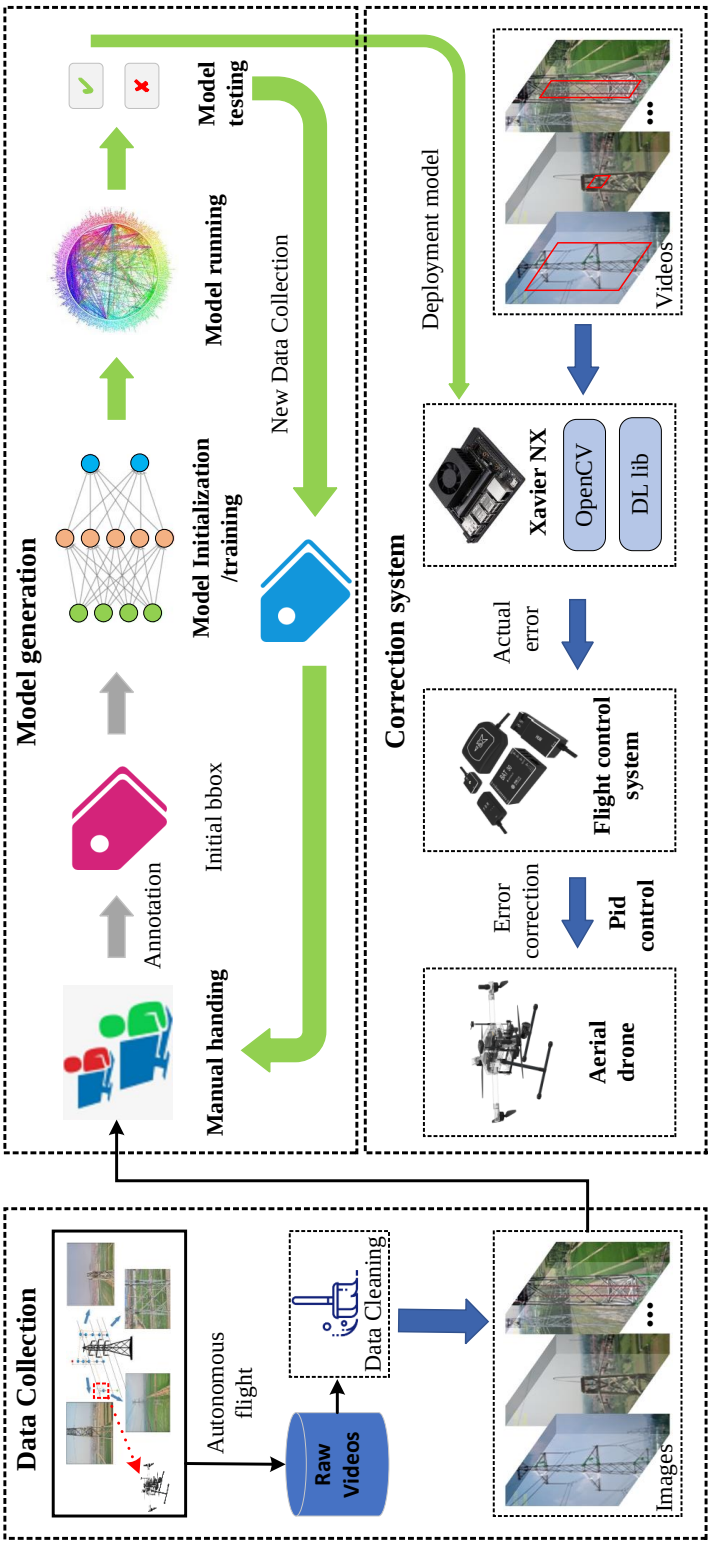


Figure 5-2: The structure of the system. The system mainly includes three parts: data collection, model generation and deployment, and correction. Data cleaning means removing useless images from the collected data. During model training, if the model is not stable enough, some new data will be collected to continue the training. The correction system is mainly based on the real-time input video stream information to calculate the position of the UAV that needs to be adjusted.

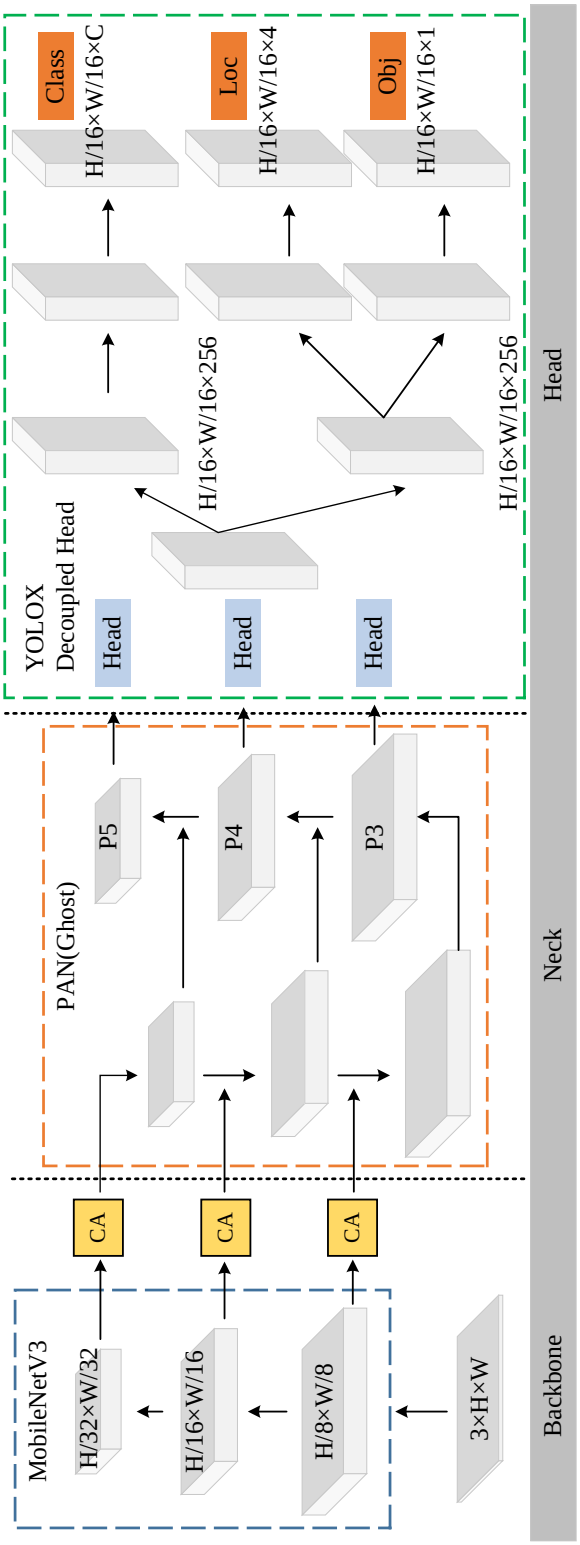


Figure 5-3: The structure of the improved YOLOX. We mainly modified the backbone and neck parts of the model, introduced MobileNetv3 and Ghost modules, and added coordinate attention after the output feature map of backbone.

area, we introduce coordinate attention[197], which embeds location information into channel attention, enabling lightweight networks. (5) To improve the accuracy of the bounding box regression, we introduce the α-DIOU[226] loss function. The optimized model is shown in Figure 5-3. We also made some application improvements based on YOLOX in another accepted article, which focuses more on accuracy because it is deployed on a server[171], while this chapter focuses more on speed because it is deployed on a UAV.

Reasonable Data Augmentation

After extensive experiments, it was found that Mosaic data augmentation had little effect on the dataset for this experiment, prolonged the training time, and greatly reduced the convergence speed. Therefore, in the experiments, we turned off Mosaic data enhancement was turned off. Reasonable data augmentation ensuress that the model is more generalizable. Considering that the light intensity and the inclination angle of the UAV are the main factors affecting this experiment, the pictures in the dataset are adjusted to a certain brightness and some small angles are rotated within plus or minus 10°. The enhanced dataset reduces the possibility of model overfitting and solves the problem of sample imbalance.

Lightweight Backbone

In real projects, an effective way to lighten the model is to replace the backbone. The backbone of YOLOX is CSPDarkNet53. In this chapter, we introduce MobileNetv3 as the backbone of YOLOX_tiny. MobileNetv3 first performs a search for coarse structures using MnasNet and then uses a reinforcement learning approach to determine the optimal configuration from a set of discrete choices. The architecture is then fine-tuned using the NetAdapt architecture, which is able to adjust underused activation channels with a relatively small drop. A novel idea of MobileNetv3 is to include squeeze-and-excitation networks (SENet) in the architecture[166]. The core idea of SENet is to model the interrelationship between feature channels in a display manner so as to automatically obtain the importance of each channel in the feature map by learning and then enhance the useful information and suppress the information that is not useful for the current task according to this importance. The SENet structure consumes a certain amount of time, but changing the channel of the expansion layer to 1/4 of the original one improves the accuracy without increasing time consumption.

MobileNetv3 is an image classification network, and to classify images, the network ends up using global pooling and a fully connected layer, which is redundant for the object detection model. To replace the backbone of YOLOX_tiny, the network structure before 32-fold downsampling is retained in the experiment and the feature maps of 8-fold downsampling, 16-fold downsampling, and 32-fold downsampling are used as the output of the backbone, which not only reduces the computation effort of the model but also satisfies the structural design of the object detection model. At the same time, to ensure the normal operation of the neck part, the number of input channels of the three neck feature maps is set to the number of output channels of the corresponding MobileNetv3 feature maps, where the number of output channels of 8-fold downsampling, 16-fold downsampling, and 32-fold downsampling

are 24, 48, and 576, respectively. Figure 5-4 provides the details.

Lightweight Neck

After the model extracts the features, an image can have many feature maps but some of the feature maps may have high similarity, leading to some redundancy in the feature maps in the neural network[227]. Consider the possibility of replacing the traditional convolution with a convolutional layer with a smaller number of output feature maps and another operation that increases redundancy and is less computationally intensive, which not only reduces redundancy to ensure accuracy but also reduces the overall computation effort of the network. Therefore, we introduce depthwise convolution and the Ghost module to replace the normal convolution in the neck. The Ghost module consists of two parts, the normal convolutional layer and the lightweight linear transform layer (depthwise convolution). The function of depthwise convolution is that the feature map of one channel is used as input, the feature map of one channel can be output, and the channels are separated from each other. Figure 5-5 presents the structure of the Ghost module. Firstly, normal convolution is used to compress the input feature maps in the channel. Then, more feature maps are obtained using depthwise convolution. Finally, the different feature maps are stitched in the channel dimension and combined into a new feature map.

Coordinate Attention

This part has been introduced in Section 4.3.6 and will not be repeated here.

Bounding Box Regression

Currently, Intersection over Union (IOU) loss function is mainly used in the field of object detection. The calculation of IOU loss function is relatively simple. The IOU loss function is obtained by dividing the intersection area of the label box and the prediction box by the combined area, taking the obtained value as the base logarithm of e, and then taking its negative value. The GIOU loss function[150] is improved by using the outer and intersecting rectangles of the label box and the prediction box for the calculation, which improves the performance of the model. A disadvantage is that GIOU degenerates into IOU when the label box is parallel to the prediction box. In addition, the convergence of GIOU and IOU is slow, while the regression is not precise enough. Only the IOU and GIOU codes are provided in the official code of YOLOX. Therefore, in this experiment, to further improve the accuracy of the model, we introduce DIOU[203] and α-DIOU loss functions. The DIOU loss function introduces the distance between the label box and the center point of the prediction box and the length of the diagonal of the minimum outer rectangle, which makes the regression more accurate. The α-DIOU loss function, shown in equation (5-2), improves the regression accuracy of the bounding box by adaptively re-weighting the losses and gradients of high and low IOU objects. The α-DIOU loss function is more favorable to the lightweight model. Therefore, in this chapter, α-DIOU loss to further improve the accuracy of the model, and here, α=3.

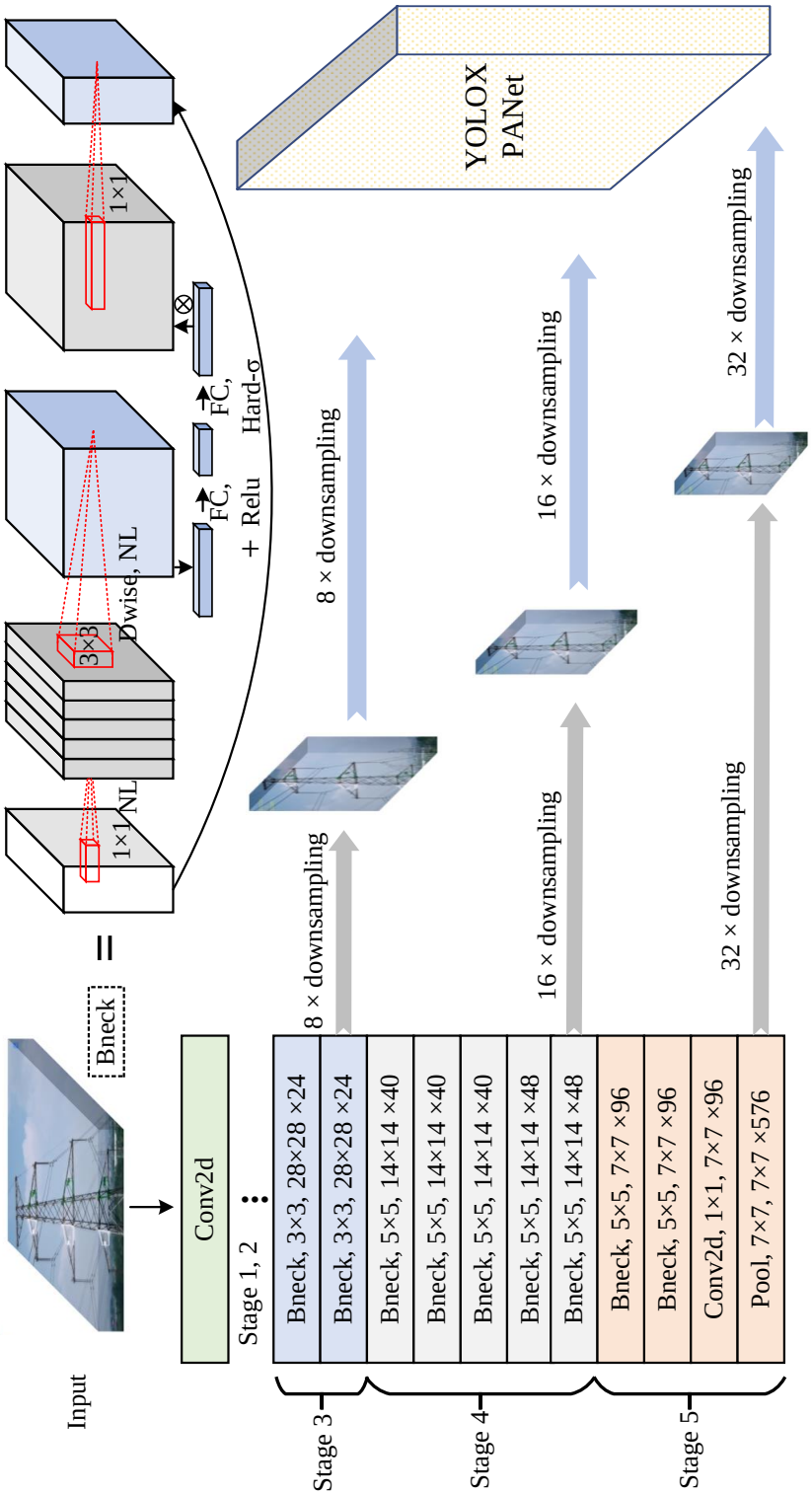


Figure 5-4: The structure of the backbone. This is the main network structure of MobileNetv3. We remove the last fully connected layer and use it as the backbone of YOLOX.

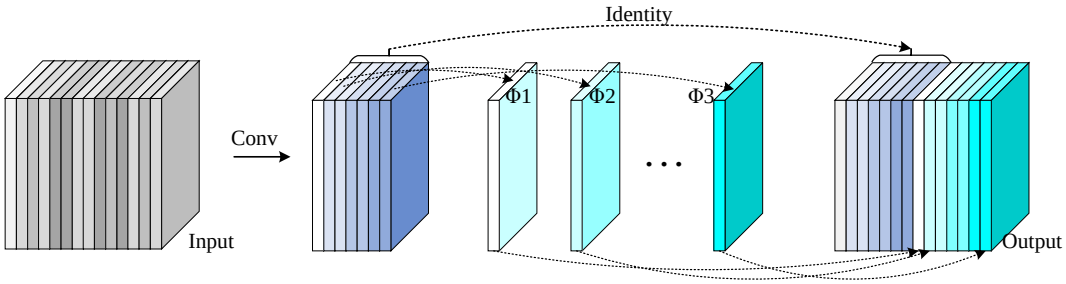


Figure 5-5: The structure of the Ghost module. Identity is equivalent to a carry operation without any convolution calculation. ϕ indicates the extraction of features using depthwise convolution.

$$L_{DIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} \tag{5-1}$$

$$L_{\alpha-DIOU} = 1 - IOU^{\alpha} + \frac{\rho^{2\alpha}(b, b^{gt})}{c^{2\alpha}} \tag{5-2}$$

5.3.3 Pixel Error to Actual Error

During the actual mission, the drone detects the corresponding target and obtains the pixel position of the target in the picture and the pixel error from the center point. However, the system needs to convert the pixel error into the actual distance error and thus adjust the position of the drone so that, as far as possible, the target is in the center of the picture.

For different targets, the zoom magnification of the lens must be determined so that the target in the picture is coordinated. According to the imaging relationship, the image distance can be calculated, as shown in equation (5-3), where I_h denotes the height of the image sensor, u is the object distance (the distance between the camera and the target), H is the actual height corresponding to the screen where the object distance is located, and v denotes the image distance.

$$\frac{I_h}{H} = \frac{v}{u} \Rightarrow v = \frac{I_h \times u}{H} \tag{5-3}$$

$$\frac{1}{f_c} = \frac{1}{u} + \frac{1}{v} \tag{5-4}$$

The Gaussian imaging equation (5-4) is substituted into the equation (5-3) to obtain equation (5-5), which is for calculating focal length.

$$f_c = \frac{I_h \times u}{H + I_h} \tag{5-5}$$

The zoom multiplier T is calculated as $T = f_c/f$, which is substituted into equation (5-5) to obtain equation (5-6).

$$T = \frac{I_h \times u}{(H + I_h) \times f} \tag{5-6}$$

In the above, f_c is the actual focal length and f is the minimum focal length, and the values of u and H at each target location are included in the mission information

sent from the ground station. Although the UAV in this experiment uses real-time kinematic (RTK)[228], due to various environmental factors, the value of u will still have some error in the actual correction, within plus or minus 15 cm. According to equation (5-6), the actual error corresponding to the pixel error can be inverted, as shown in equations (5-7) and (5-8).

$$\Delta H = \frac{\frac{\Delta y}{C_y} \times I_h \times u}{f \times T} - I_h \tag{5-7}$$

$$\Delta W = \frac{\frac{\Delta x}{C_x} \times I_w \times u}{f \times T} - I_w \tag{5-8}$$

Δy and Δx represent the pixel difference between the height and the width of the target from the image centroid, respectively, and C_y and C_x represent the pixel values of the actual image height and width, respectively. I_w denotes the width of the image sensor, and ΔH and ΔW denote the actual error corresponding to the pixel error in the vertical and horizontal directions. In the actual experiment, $C_y = 768$, $C_x = 1024$, $I_h = 5.65mm$, $I_w = 9.35mm$, and $f = 6.7mm$. In the calculation, the units should be consistent. Otherwise, the calculation results will have large errors.

5.3.4 UAV Autonomous Correction Inspection System

This section briefly introduces the entire process of UAV power inspection. The UAV consists of three main parts: a path planning module (PPM), a vision-based correction module (VCM), and the servo control of the UAV module (SCM). The PPM is realized by the ground station, which automatically plans an autonomous flight path that conforms to the tower according to the relevant information of the electric tower to be inspected. The VCM acts in the arrival stage of the aircraft position point, mainly for the problem that the target to be inspected deviates from the center of the image. The SCM is the basis of the entire inspection system[195].

Path Planning Module

In the PPM, each key location of the autonomous flight path is generated by the base information of the target tower to be inspected, which includes the latitude, the longitude, the direction, the tower height, and the type of the tower. Each type of tower has its fixed properties, such as a fixed crossbeam length and the height difference between the crossbeams, taking the tower shown in Figure 5-6 as an example (Since it is the same task, so the path planning is consistent with the path planning in the section 4.3.2). After the coordinates of the tower center and its direction are known, the expression of the linear equation of points 2 and 8 in the world coordinate system can be determined perpendicular to the direction and through the tower center coordinates. The specific latitude and longitude information of points 2 and 8 can be determined by combining the maximum crossbeam length and the safety distance. The safety distance is to prevent the high voltage from affecting the drone magnetometer and the GPS. Combined with the tower height, the specific 3D information of the point can be obtained. The target heading of point 2 is perpendicular to the direction of the tower toward the side of the tower, while the

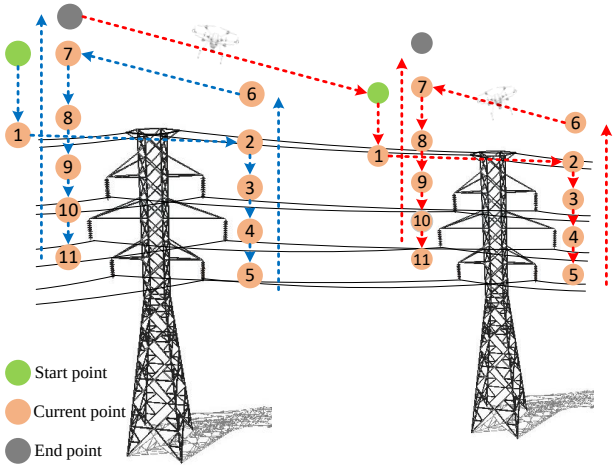


Figure 5-6: Path planning diagram. The orange dots in the figure indicate the mission points of the UAV.

specific location information for points 3, 4 and 5 can be obtained by subtracting the height difference of the crossbeam from the information for point 2. Points 9, 10, and 11 can be obtained in the same way according to point 8. As for points 6 and 7, they are determined by adding a certain safety distance on top of points 2 and 8, respectively.

Vision-based Correction Module

The autonomous flight path planned according to the tower information will enable the UAV to collect the image information of each target object in the tower well. The waypoints and path planning positions are precise, so the error depends on the accuracy of the drone positioning. The error in the position of the RTK into the fixed solution is very small, and it is almost guaranteed to reach a very precise position every time. However, the high voltage of the towers, the thickness of the clouds can cause some signal interference, the number of satellites in the sky at different times can have an effect, and excessive wind speed can also cause the position of the UAV to deviate, thus causing the target object to deviate from the image center and even incomplete data collection, this will have a great impact on the later inspection work. While VCM can correct the position of the UAV based on the information of the detected target in the image, and this process is shown in Figure 5-7.

From equations (5-7) and (5-8), the pixel error can be converted into the actual distance error, which is the actual error distance of the target from the Y-axis and Z-axis of the UAV. In UAS, the input position target is usually represented in the form of latitude and longitude. therefore, we also need to convert the position error into the latitude and longitude coordinate system. The conversion is shown in equations (5-9) and (5-10), where M and N is the latitude and longitude information, respectively, of the current location of the UAV; OW is the distance of the object from the Y-axis of the UAV; a indicates the direction perpendicular to the current heading of the UAV, which is related to whether the target object is located on the left or the right side of the UAV; p and q are the relationship coefficients between the actual distance and the latitude and the longitude, respectively, determined

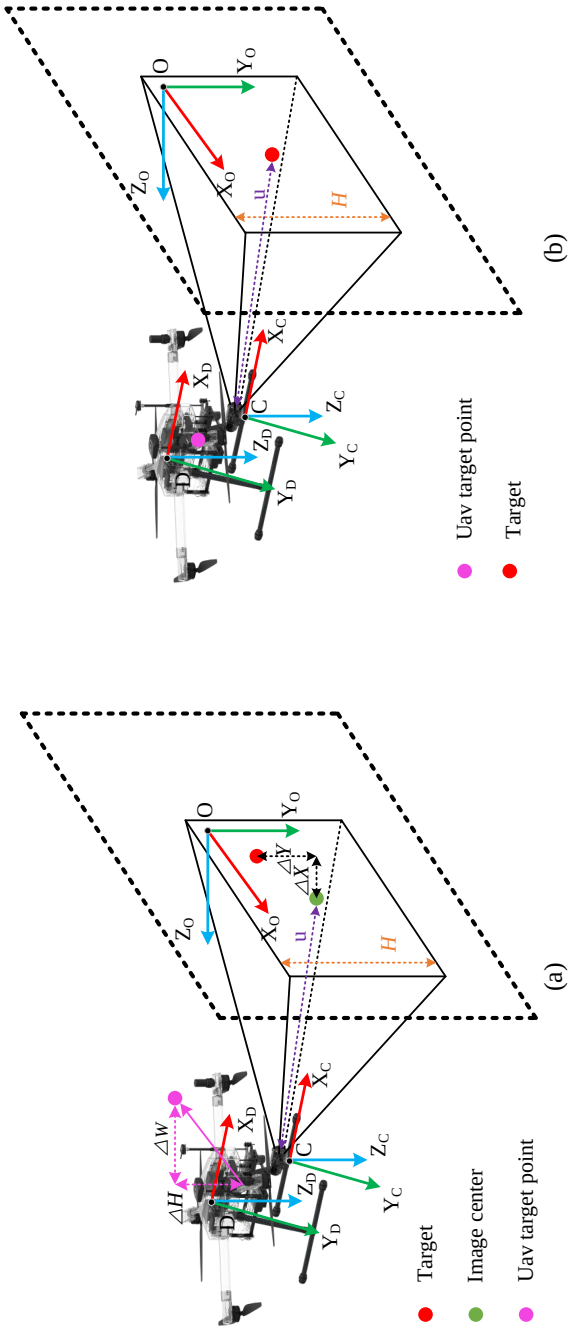


Figure 5-7: The correction process of the drone. (a) The position of the UAV before it corrects the deflection, at which time the target is in the top-left corner of the picture. (b) The position of the UAV after the correction; at this time, the target is in the center of the picture. ΔX and ΔY represent the pixel error of the target from the center of the picture, ΔH and ΔW represent the actual corrective distance of the UAV. As per the image sensor, I_w and I_h are 7.35mm and 5.65 mm, respectively, and the values of the object distance u and the actual height H corresponding to where the object distance is located can be obtained from the mission information sent from the ground station.

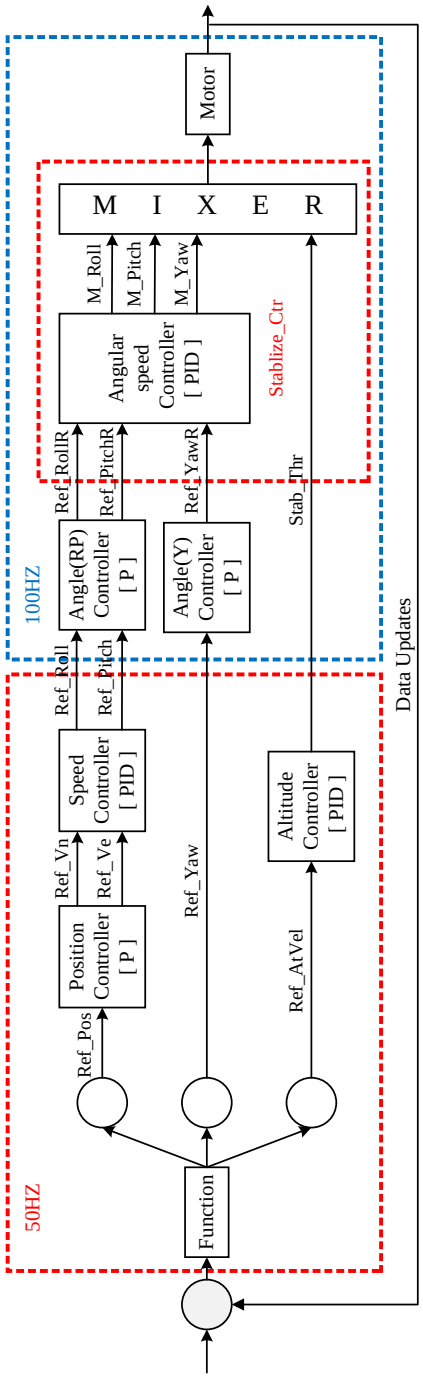


Figure 5-8: The structure of the UAV control. The whole control system is designed with a series structure. Because the update frequency of various sensor data is different, the design of sub-controllers is completed at different frequencies.

by the current latitude and longitude; and X and Y are the corrected longitude and latitude information, respectively, of the target location. For the actual error distance in the Z-axis direction, it is only necessary to modify it on the basis of the current target height. In this way, the flight point position of the UAV is adjusted according to the pixel error of the target recognition result, which can finally ensure that the target is in the center of the acquired image, thus ensuring the effectiveness of data acquisition.

$$X = M + (OW * \frac{cos(a)}{p}) \tag{5-9}$$

$$Y = N + (OW * \frac{sin(a)}{q}) \tag{5-10}$$

Servo Control of UAV Module

UAS is used to perform the entire inspection, and the UAV control system adopts a string control structure, including position control, speed control, attitude control, and stability control, as shown in Figure 5-8. Since the results of PPM and SCM calculations are related to the 3D position, the focus is on the two upper layers of the UAV: position control and altitude control. We apply the PD control algorithm in the upper-level control, using the task points calculated in PPM and SCM as target points for control. In position control, we calculate the target information required for speed control by combining the target position information, latitude, and longitude with the current state. We calculate the error between longitude and latitude separately and convert this error into distance error to obtain the actual error distance in the N, E direction. Considering that the distance between the target position and the current position may be too large, to ensure control accuracy, it is also necessary to limit the error distance processing so that when the distance error is small, the UAV can also respond in time. In this case, the calculation process of the velocity target value in the N direction, for example, is shown in equation (5-11).

$$V_{N-ref} = K_{p1} \cdot Sat(x_{ref} - x) + K_{d1} \cdot \frac{d(Sat(x_{ref} - x))}{dt} \tag{5-11}$$

Where x represents longitude, K_{p1} and K_{d1} are the control coefficients, and the $Sat(x)$ function is the saturation function to limit the excessive distance error. Similarly, the target value in the E direction is shown in equation (5-12), where y is the latitude.

$$V_{E-ref} = K_{p1} \cdot Sat(y_{ref} - y) + K_{d1} \cdot \frac{d(Sat(y_{ref} - y))}{dt} \tag{5-12}$$

For height control, we still use PD control. We can calculate the control amount of height direction by combining the target value of height with the current height, as shown in equation (5-13).

$$u_{Thr} = K_{p2} \cdot Sat(h_{ref} - h) + K_{d2} \cdot \frac{d(Sat(h_{ref} - h))}{dt} \tag{5-13}$$

5.4 Experiments

In this section, first, we introduce the dataset used in this chapter, the evaluation metrics of the model, and the training method. Then, we conduct ablation experiments to verify the performance of the model, followed by actual flight verification. Finally, a comparison is made with other high-voltage power transmission line inspection systems.

5.4.1 Dataset Introduction

In this study, the dataset was obtained mainly using self-developed UAV flights. After receiving the task from the ground station, the UAV performs the photo operation according to the location points, as shown Fig. 10. To build the dataset for this experiment, the UAV autonomously recorded and photographed during the operation. The dataset of this experiment mainly includes electric towers, voltage equalization rings, insulators, and overhanging wire clamps. The experimental site is in Xuzhou City, Jiangsu Province, China, and the experimental equipment is the UAV shown in Figure 5-9. For this experiment, about 4000 images were collected from the dataset.

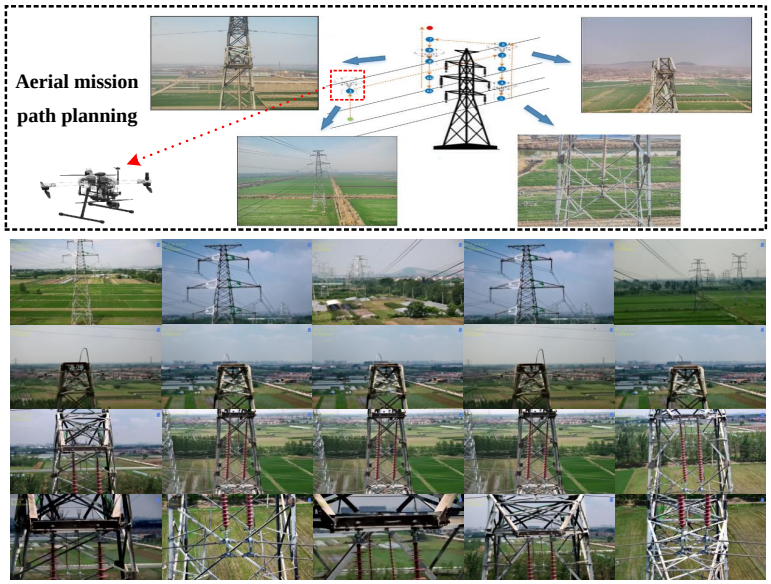


Figure 5-9: Image of part of the dataset.

5.4.2 Evaluation Metrics

This part has been introduced in Section 4.4.2 and will not be repeated here.

Speed is another important metric to measure the object detection model. Only fast enough to achieve real-time processing, this metric is called frames per second (FPS), which is the number of images processed per second.

5.4.3 Model Training

The hardware platform for this experimental training model is as follows: the GPU is GeForce RTX3090, the CPU is Intel(R) Core(TM) i9-12900K, the video memory

is 24G, the OS is Windows 11, the application development language is Python, and the deep learning framework is Pytorch. For end-to-end training of the model in the experiments, we used a stochastic gradient descent method. The parameters of model training are set as follows: the size of the batch is 32, the initial learning rate is set to 0.01, the weight decay is set to 0.0005, the size of the input network image is 768×1024, no Mosaic data enhancement is used, and the L1 loss function is increased from the beginning of training. Other parameters are basically the same as the training parameters of YOLOX.

5.4.4 Comparison of Models

YOLOX is currently one of the better balanced object detectors in terms of speed and accuracy. YOLOX has six different versions: YOLOX_s, YOLOX_m, YOLOX_l, YOLOX_x, YOLOX_tiny, and YOLOX_nano. YOLOX_tiny is a lightweight model, suitable for robotics, UAVs, and other airborne embedded deployments, so we chose YOLOX_tiny as the baseline.

To verify the feasibility of the improved method, we conducted a series of ablation experiments, as shown in Figure 5-10 and Figure 5-11. For the backbone of the model, in the experiments, two smaller backbones in image classification were selected to replace CSPDarkNet53 in YOLOX_tiny, ShuffleNetv2 and MobileNetv3, where MobileNetv3 is the small version. Since the $mAP\_0.5$ of YOLOX_tiny is overfitted after training more than 200 epochs, and the accuracy of $mAP\_0.5{:}0.95$ also tends to be flat, only 300 epochs are trained. ShuffleNetv2 and MobileNetv3 converged slower than CSPDarkNet53, so with ShuffleNetv2 and MobileNetv3 as the backbone, the model was trained for 500 epochs. The experimental results show that the network model with ShuffleNetv2 as the backbone runs much slower, despite the improvement in accuracy over the baseline. The network model with MobileNetv3 as the backbone is 0.006 and 0.003 percentage points lower than the baseline in $mAP\_0.5$ and $mAP\_0.5{:}0.95$ metrics, respectively, but the operation speed is improved by about 20 FPS, so we selected MobileNetv3 as the backbone of the baseline.

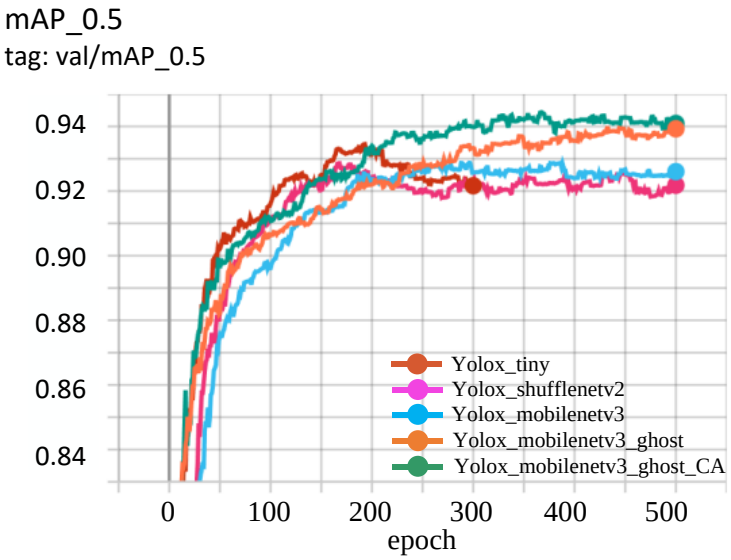


Figure 5-10: The $mAP\_0.5$ data curve under different models.

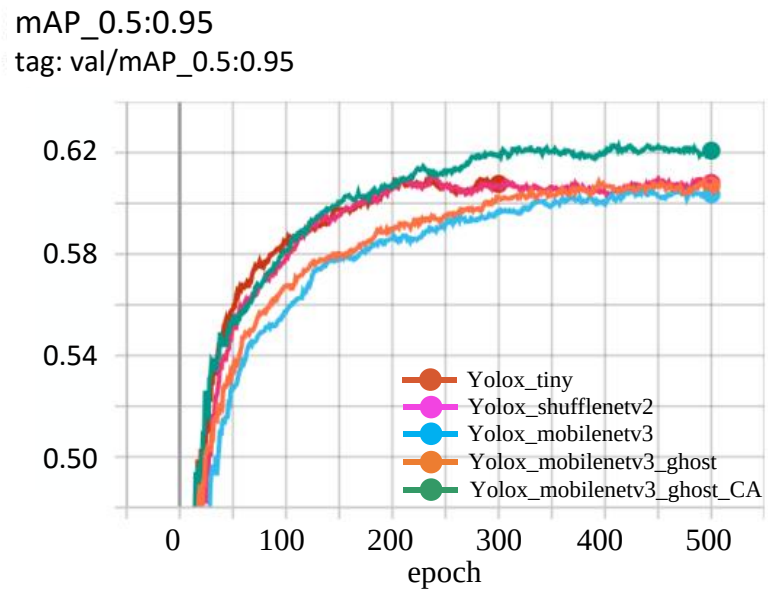


Figure 5-11: The $mAP\_0.5:0.95$ data curve under different models.

Based on YOLOX_mobilenetv3, Ghost module and depthwise convolution are introduced in the neck to further verify the performance of the model. The experimental results show that the improved network model is basically the same than the original model in terms of inference speed, but improves 0.006 and 0.004 percentage points in $mAP\_0.5$ and $mAP\_0.5:0.95$ metrics, respectively, while the number of parameters of the model is greatly reduced, almost reducing the number of parameters by nearly half. Then the attention mechanisms such as SENet, CBAM and CA are added to improve the accuracy of the model respectively, and the experimental results show that the performance of adding CA to the model is optimal. The model improved 0.003 and 0.002 percentage points over SENet and CBAM, respectively, in the $mAP\_0.5$ metric, and 0.008 and 0.006 percentage points in the $mAP\_0.5:0.95$ metric. YOLOX_mobilenetv3_ghost_CA compared to YOLOX_tiny, the number of parameters is reduced by 2M and Gflops by 5.92. $mAP\_0.5$ and $mAP\_0.5:0.95$ are improved by 0.1 and 0.14 percentage points respectively, while the running speed is increased by about 15 FPS. The details are shown in Table 5-1.

On the basis of the above, we introduced α-GIOU, DIOU, and α-DIOU to improve the accuracy of the bounding box regression. $mAP\_0.5$ and $mAP\_0.5:0.95$ curves are shown in Figure 5-12 and Figure 5-13. As per the figure, the performance of α-DIOU is optimal. α-DIOU improves the $AP\_0.5:0.95$ metric by about 1 percentage point compared with IOU and GIOU.

To further validate the performance of the model, we compared the improved model with other models, as shown in Table 5-2, where FPS is the speed of the model deployed on Nvidia NX. The results show that the improved model reduces by 4.98M, 2.75M, 0.67M, and 2.01M in terms of the number of parameters compared to YOLOv3_tiny, YOLOv4_tiny, Efficientdet-d0, and YOLOX_tiny, respectively. In the $mAP\_0.5:0.95$ metric, although it is 0.003 lower than Efficientdet-d0, it is more accurate than any other model and the improved model is also fast (2.8 times faster than Efficientdet-d0) and achieves almost the same speed as YOLOX_nano.

Table 5-1: Comparison of ablation experiments

Method	**Size**	**Par**	**Gflops**	$\boldsymbol{mAP_{0.5}}$(%)	$\boldsymbol{mAP_{0.5:0.95}}$(%)	$\boldsymbol{AP_S}$	$\boldsymbol{AP_M}$	$\boldsymbol{AP_L}$	**FPS**
YOLOX_tiny	768×1024	5.03M	29.26	0.932	0.608	0.738	0.574	0.617	90
YOLOX_shufflenetv2	768×1024	14.7M	51.34	0.929	0.613	0.758	0.617	0.615	74
YOLOX_mobilenetv3	768×1024	5.78M	26.32	0.926	0.605	0.699	0.617	0.609	**110**
YOLOX_mobilenetv3_ghost	768×1024	**2.98M**	**23.32**	0.939	0.609	0.732	0.616	0.616	109
YOLOX_mobilenetv3_ghost_SENet	768×1024	2.99M	24.33	0.939	0.614	0.745	0.622	0.624	106
YOLOX_mobilenetv3_ghost_CBAM	768×1024	2.99M	24.33	0.640	0.616	0.752	0.628	0.631	106
YOLOX_mobilenetv3_ghost_CA	768×1024	3.02M	23.34	**0.942**	**0.622**	**0.764**	**0.642**	**0.643**	105

This table compares the performance of different backbones and different attention mechanisms in YOLOX_tiny, where blackened fonts are the best performers, but all together YOLOX_mobilenetv3_ghost_CA is the best choice. Par is the meaning of the parameter.

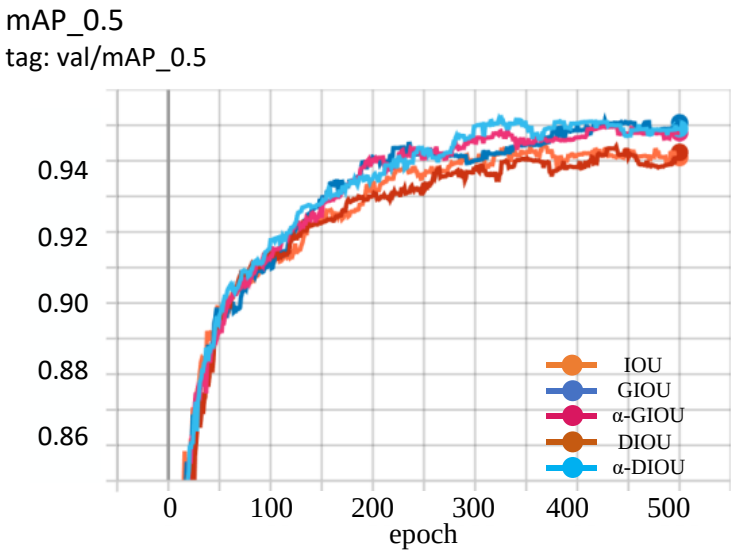


Figure 5-12: The mAP_0.5 data curve Under different loss functions.

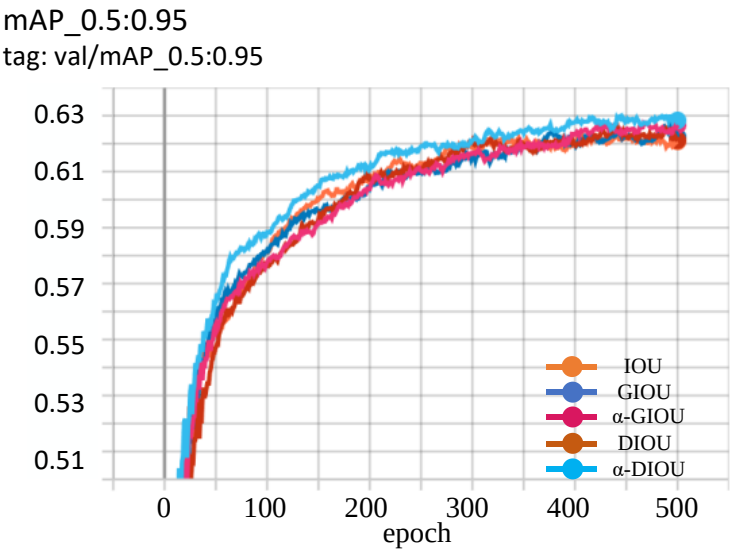


Figure 5-13: The mAP_0.5:0.95 data curve Under different loss functions.

Table 5-2: Performance comparison of different models

Methods	**Par**	**mAP_0.5**	**mAP_0.5:0.95**	**FPS**
YOLOv3_tiny	8M	0.865	0.573	50
YOLOv4_tiny	5.77M	0.926	0.594	48
Efficientdet-d0	3.69M	0.948	**0.632**	20
YOLOX_nano	**0.9M**	0.728	0.436	**60**
YOLOX_tiny	5.03M	0.932	0.608	50
Ours	3.02M	**0.951**	0.629	56

This table compares our improved model with other models, where the bolded font is the best performer, but in terms of speed and accuracy combined, the model we provide is the best.

5.4.5 Test of Actual Flight

To verify the effectiveness of the proposed correction system, it is necessary to carry out an actual flight in a real environment. In the experiment, we used the team's self-developed inspection UAV as the validation platform. The flight controller motherboard is designed with an integrated dual processor, where the STM32F427ZIT6 based on the CORTEX-M4 core is used to receive and process data from the network RTK module and to perform calculations to provide high accuracy position information for the inspection drone; the STM32H753VIT6, based on the CORTEX-M7 core, runs the inspection drone control program, which receives and processes data from the individual sensor units for data fusion and ensures stable flight of the inspection drone. The whole system integration board uses dual processors for simultaneous operation and fuses discrete functional units designed into the integrated main board. The main integrated units are the data transmission unit DTU, the data recording unit FDR, the status indicator LED, the CAN signal hub and the power supply voltage conversion unit. For safety reasons, the verification object chosen was a linear pole tower in an unoccupied and open area in Xuzhou City, Jiangsu Province, China, and was approved by the relevant authorities for multiple flight verification. The experimental procedure is shown in Figure 5-14.

Figure 5-14: Actual experimental environment.

The detailed settings and parameters of the drone are as follows: size 600 × 600 × 450 mm, total take-off weight 6.85kg, battery capacity 22000 mAh, flight time about 42 min, ascent speed 3 m/s and descent speed 2 m/s. The position controller is implemented through P control with the parameter $K_{pp} = 0.076$. The speed controller is implemented by a PID controller with the parameters $K_{pv} = 0.076$, $K_{iv} = 0.016$, and $K_{dv} = 0.02$, respectively. In the attitude controller, roll and pitch attitude are controlled by P of $K_{pa} = 600$ and the yaw direction is controlled by P of $K_{py} = 1500$. The height controller is implemented using the PID method, where $K_{ph} = 300$, $K_{ih} = 200$, and $K_{dh} = 10$. Finally, the underlying controller is implemented by a PID control with parameters $K_{ps} = 60$, $K_{is} = 160$, and $K_{ds} = 3$. The experimental results will be discussed and analyzed next.

Data of The Flight

The UAV took off from an open area near the pole tower and completed the in-

spection operation according to the process described in Section 5.3.4. For analysis, the actual latitude and longitude of the UAV have been converted into the actual distance in the (N, E, D) coordinate system. As shown in Figure 5-15, the UAV took off from (0,0,0), inspected the pole tower with coordinates (-117, 103, 46), and returned to the starting point after the inspection was completed. The detailed trajectory of the inspection process and the actual target point location are shown in Figure 5-16, where the red dots are the actual target location information of the UAV. It is worth noting that the target points B to L in the figure correspond to points 1 to 11 in Fig. 7, which means that the UAV can work exactly as the expected flight path. Figure 5-17 presents the velocity target value during the process with its tracking.

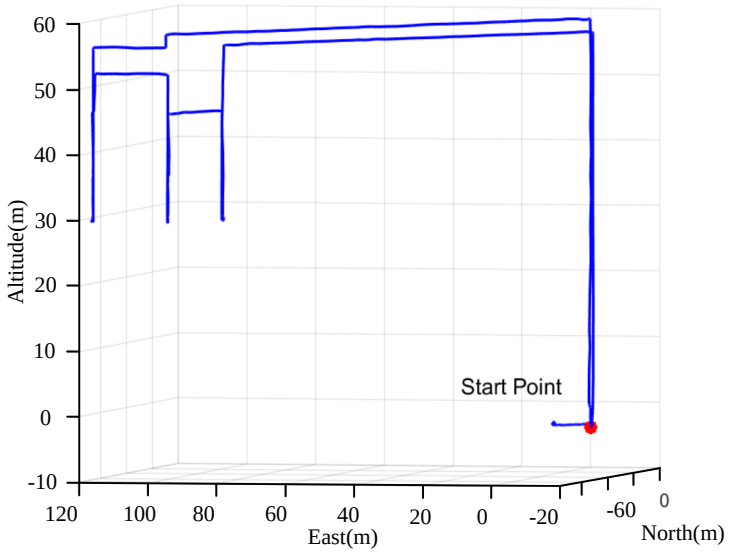


Figure 5-15: The complete flight path of the UAV.

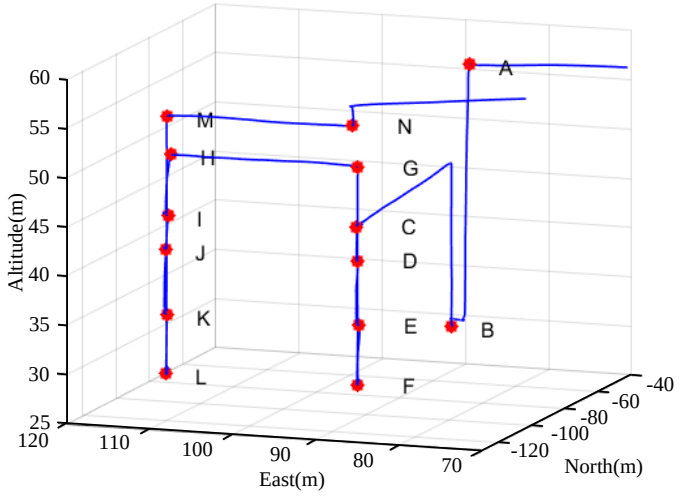


Figure 5-16: Mission trajectory of the UAV. The red dots are mission points for UAVs.

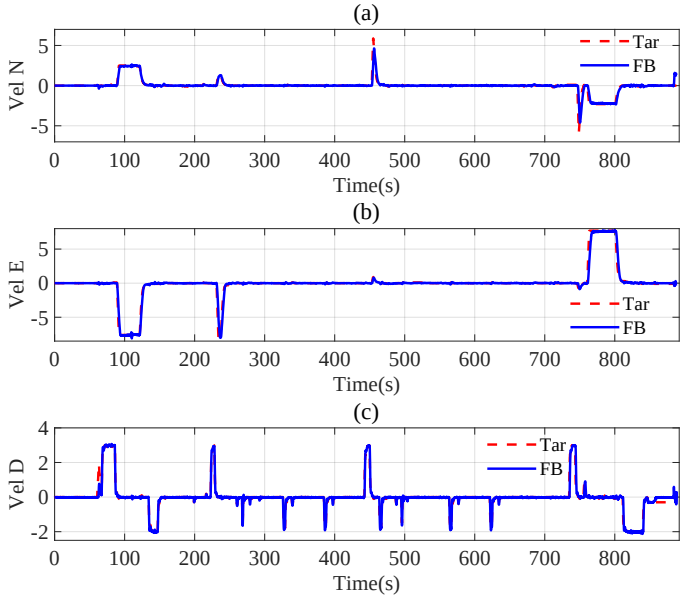


Figure 5-17: Results of speed control. The dashed red line is the target velocity, and the solid blue line is the actual velocity. (a), (b), and (c) correspond to the tracking effect of the UAV in the N, E, and D directions, respectively.

From the description in Section 5.3.4, it is clear that the speed control of the UAV, as the middle part of the whole control system, has a tracking accuracy that can well characterize the effectiveness of the UAV flight. Figure 5-17(a) shows the target and the actual values of the velocity in the N direction, Figure 5-17(b) provides information in the E direction, and Figure 5-17(c) shows the target and actual values of the velocity in the vertical direction of the UAV. These results indicats that the UAS can have good tracking performance and can follow the planned target trajectory exactly.

During the whole flight, the resolution size of the image was 768×1024. Figure 5-18 shows the result of object detection. Figure 5-18(a) shows the error of the horizontal coordinate of the target rectangular box from the center of the picture, and Figure 5-18(b) shows the error of the vertical coordinate of the target rectangular box from the center of the picture. The values -512 and -384 are intended to be distinguished from the 0 value, specifically refers to the case where no target is detected in the image, and the 0 value refers to the case where the UAV does not perform a correction of deflection. Figure 5-18(c) and Figure 5-18(d) show the width and the height of the target rectangular box, respectively. To view the details of the correction more specifically, the object detection results and the UAV target latitude and longitude are shown in Figure 5-19 and Figure 5-20 for the data from 320 s to 360 s, respectively. As can be seen in Figure 5-19, at 330 s, the object detection system detects the presence of the target, and after calculating the desired target point location based on equations (5-9) and (5-10), the control system substitutes the newly generated point location as the new target value for the calculation, and the time points of this process can correspond exactly to the time points of the target latitude and longitude as well as the target height change process in Figure 5-20. And it can be seen in Figure 5-19(a), (b) that the center point of the target rectan-

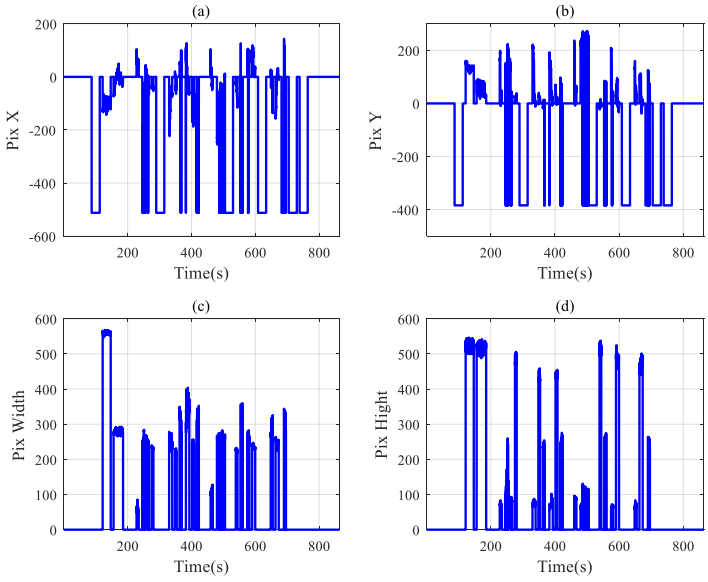


Figure 5-18: (a) and (b) are the pixel difference between the centre of the target and the centre of the image. (c) and (d) are the pixel values of the height and width of the target external rectangular box.

gular box keeps converging from 330 s to 338 s. The experimental results are fully able to show that the proposed correction system is actually effective. There is no object detection between 340 s and 350 s so the value is 0.

Result of Image Correction

Figure 5-21 shows the actual corrective deflection results. The center of the viewpoint of the UAV is clearly getting closer to each target step by step, and although there is still some error, the problem of the target deviating from the center of the picture has basically been resolved. In Figure 5-22, the initial position of the target is closer to the center, which is the main reason why the correction is not too obvious. An object detection model that inputs a graph, outputs a tensor, and then goes on to parse the tensor, which contains information about the box position, the probability of belonging to each category, etc. It is just that in the deployment, this part of the information is not shown in the video. In general, there is usually only one target at the waypoint location, and if other targets have an impact on the model, we choose the target with the highest probability value for position adjustment.

5.4.6 Comparison of Systems

We also compared our system with other system solutions for high-voltage power transmission line inspection, as shown in Table 5-3. Although Yang *et al.*[193] and Nguyen *et al.*[192] applied deep learning to high-voltage power transmission line inspection system to improve the efficiency and accuracy of detection, it does not have the function of path planning and autonomous data collection. The high-voltage power transmission line inspection system proposed by Li *et al.*[187] and Guan *et*

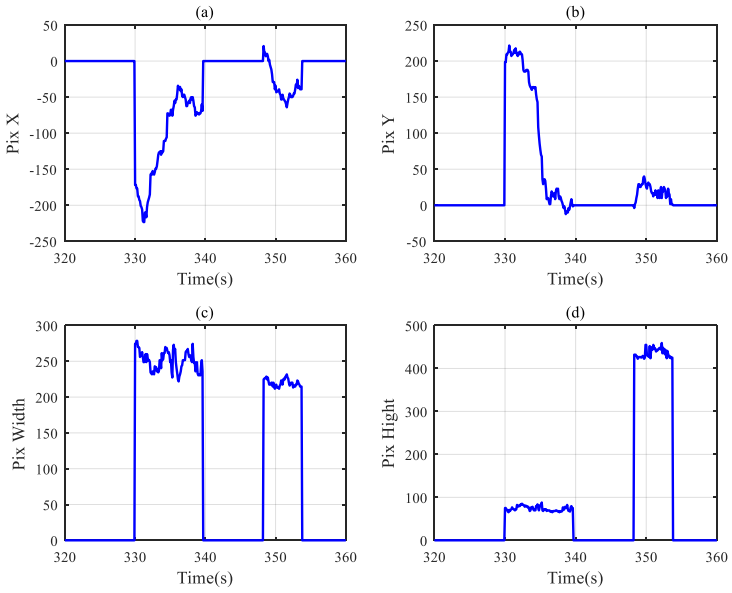


Figure 5-19: Local zoom information of object detection results. (a) and (b) are the pixel difference between the centre of the target and the centre of the image. (c) and (d) are the pixel values of the height and width of the target external rectangular box.

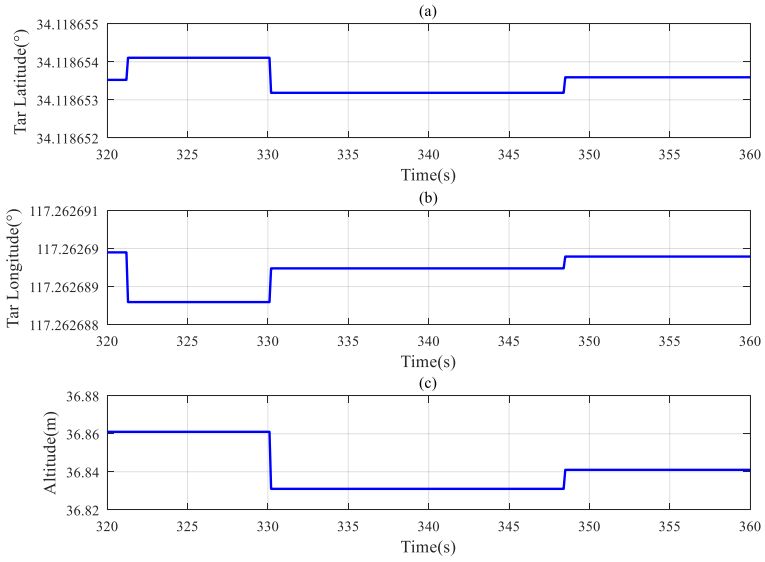


Figure 5-20: Amplification information of UAV longitude, latitude and altitude target values.

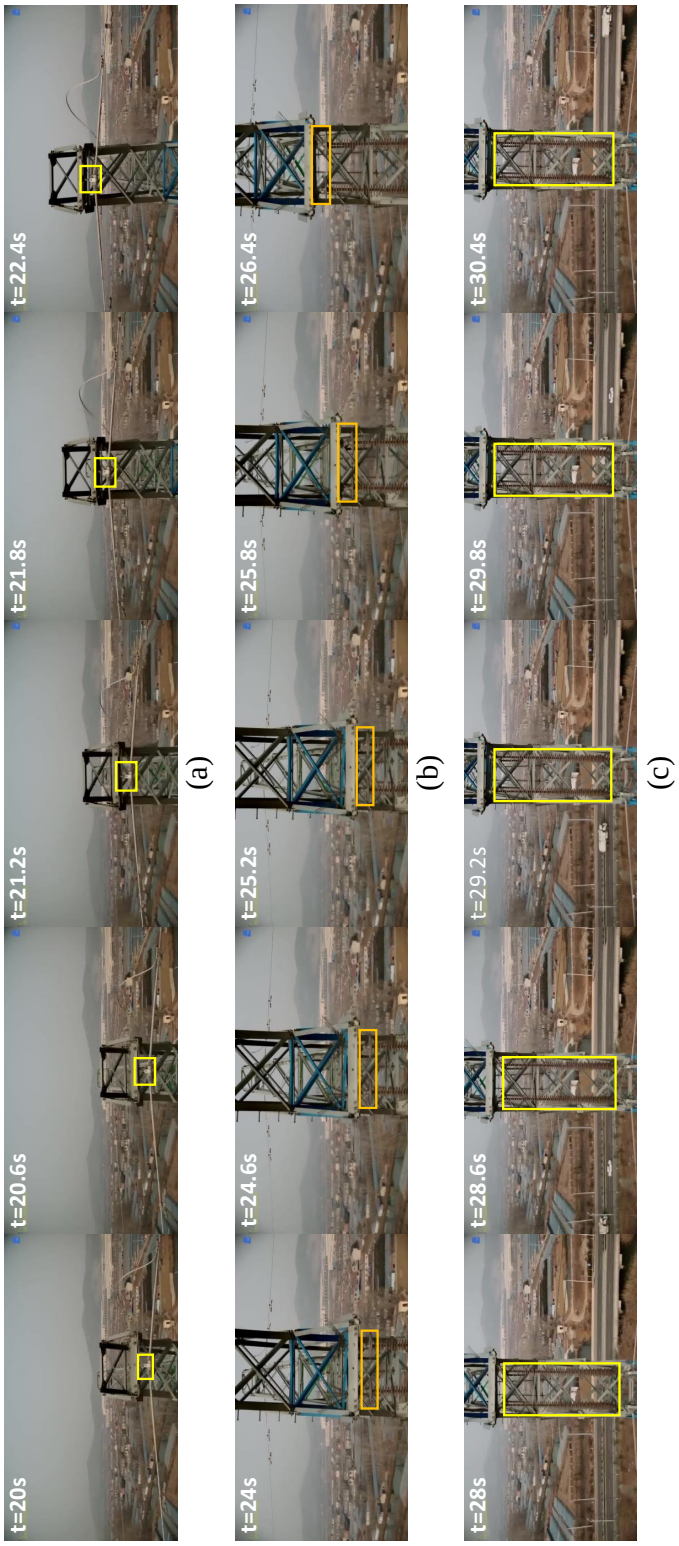


Figure 5-21: (a) Overhanging wire clamp correction process. (b) Equalizing ring correction process. (c) Insulator correction process. The rectangular box in the figure is the result of object detection and also indicates the position of the target in the image.

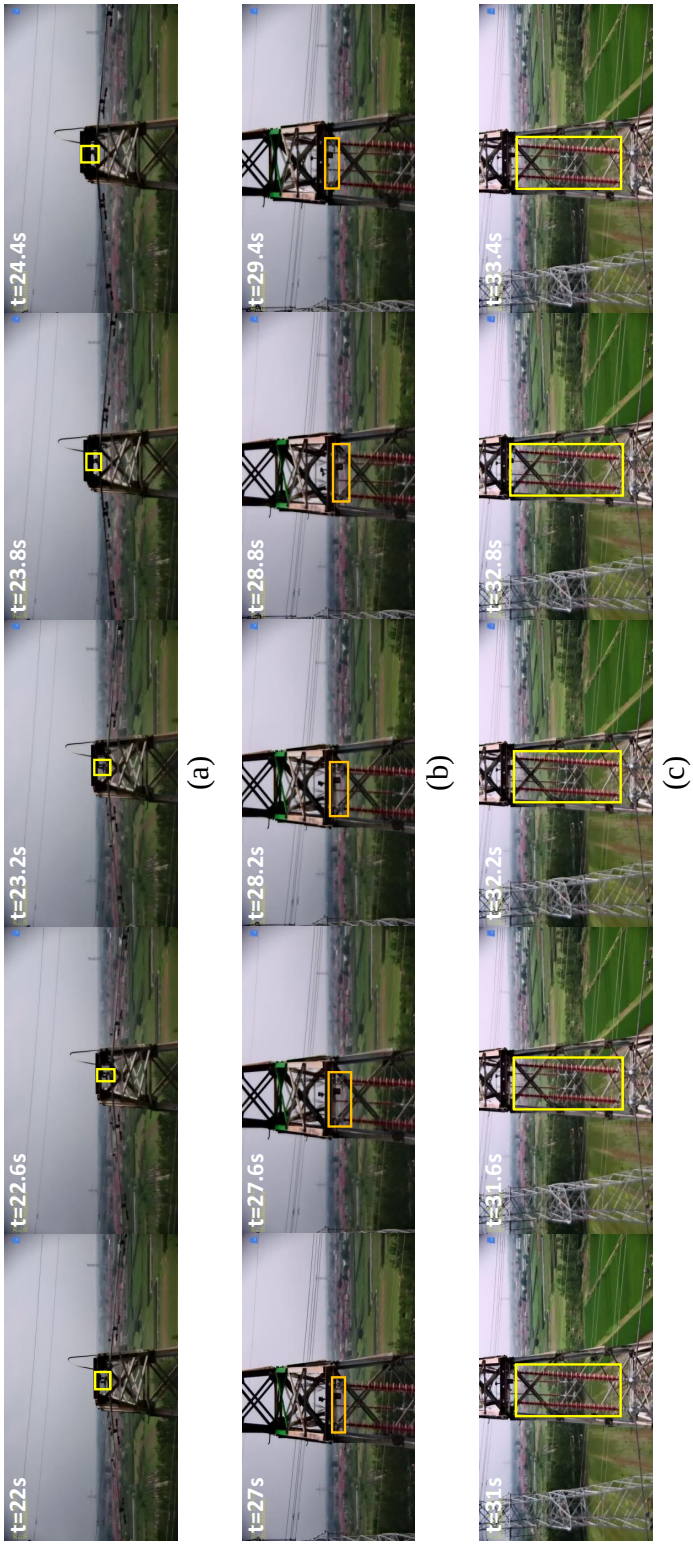


Figure 5-22: (a) Overhanging wire clamp correction process. (b) Equalizing ring correction process. (c) Insulator correction process. The rectangular box in the figure is the result of object detection and also indicates the position of the target in the image.

Table 5-3: Comparison of high-voltage power transmission line inspection systems

Systems	**Intelligent detection**	**Autonomous data collection**	**Path planning**	**Autonomous correction**
193	✓	×	×	×
192	✓	×	×	×
187	✓	✓	✓	×
188	✓	✓	✓	×
Ours	✓	✓	✓	✓

al.[188] has almost all the functions required for high-voltage power transmission line inspection, but ignores the position error of the UAV at the mission point, which may cause inaccurate data collection. Our proposed UAV high-voltage power transmission line autonomous correction inspection system not only achieves autonomous path planning, but also solves the problem of inaccurate data collection by UAV at the mission point.

5.5 Discussion

The design of this system mainly considers the practical application of the autonomous operation of a UAV. The lightweight model can be easily deployed on various embedded devices to achieve real-time detection. The small calculation size, the requirement of few parameters, and fast detection ensure that the improved YOLOX model is extremely competitive for embedded device deployment. In summary, the system proposed in this chapter has the following advantages: (1) It can detect the targets in the video stream in real time. (2) The improved YOLOX model is small and easy to deploy in embedded devices, which greatly reduces the hardware cost and is useful for practical applications. (3) The position of the target can be adjusted in real time so that, as far as possible, the target is in the center of the image, ensuring the effectiveness of the collected data.

Test video can be viewed at https://www.youtube.com/watch?v=rCiiynvFGDE and the video of actual flight can be viewed at https://www.youtube.com/watch?v=bLG6XtB2xS8. From the video, we can see that the corrective effect of the UAV is good but the system is not good at detecting the voltage equalization rings below the mission point. This may be because the number of datasets belonging to this category is small, the background at the time of detection is relatively complex, and the light intensity also produces some influence, which is a drawback of this chapter. Next, we may need to collect more data and perform some targeted data enhancement to improve the generalization of the model.

At present, there are many object detection models applied to UAS, but most of them are for real-time monitoring, and there is relatively little work on determining object positions for correction with the help of object detection models. Of course, in the case of rich datasets, we believe that the initial YOLOX_tiny can also be used to complete the work well. However, our YOLOX_tiny has improved in terms of precision and speed. So the improved YOLOX_tiny has higher generalization with limited computational resources.

With regard to energy consumption, we have set several strategies. For example, the drone consumes the least amount of energy when hovering, so we can reduce some of the energy consumption by only adjusting the camera angle and letting the drone hover when shooting the same group of three targets. Besides, when the UAV returns to descend, the UAV is made to land in sections, initially at a very fast speed, and near the ground, at a low speed, to ensure safety and at the same time effectively reduce energy consumption. In addition, the project is a collaborative project with the electric power company and allows to get specific information about the electricity towers (including tower type, latitude, and longitude information). The ground station generates waypoints autonomously and is able to generate the target heading angle (yaw) at the same time, based on the target position. Thanks

to the RTK system on board, the accuracy of the position and heading angle is guaranteed. During a mission, the targets will all be directly in front of the UAV, with a position accuracy within 15 cm and an angle accuracy of plus or minus 1 degree.

5.6 Conclusion and future work

To realize high-precision autonomous inspection of high-voltage power transmission line by UAVs, in this chapter, we designed a real-time correction system based on the YOLOX network model. On the basis of the UAV autonomous operation task, the corresponding dataset was built. To lighten the YOLOX network, the backbone was replaced with MobileNetv3 and the Ghost module was introduced to reduce the number of parameters and the amount of computation. To compensate for the accuracy problem caused by the lightweight network, CA and α-DIOU loss function were introduced to improve the performance of the model. The running speed is greatly improved, while model accuracy is guaranteed. This system also implements the conversion of pixel error to spatial position. It enables the UAV to correct its position in time according to the pixel error. In conclusion, this chapter designs the completed UAV correction inspection system, including vision module, control module, and some strategies. The experiment proves that this system effectively solves the problem of the target deviates from the center of the picture when the UAV takes pictures during a high-altitude inspection.

At present, we have completed high-quality data collection only for the linear tower types. We still need to design reasonable path planning and correction strategies for the other tower types. Once high-quality data has been collected, we plan to use a high-accuracy object detection model to detect various defects in the towers, such as tilted insulator strings, shifted equalization rings, and defective locking pins. Intelligent inspection replaces traditional manual inspection, which will greatly reduce labor costs and improve inspection efficiency. We hope that this research will drive progress in the electrical inspection industry.

6 Conclusions

6.1 Conclusion

In this thesis, we have attempted to design an intelligent inspection system for transmission lines, and we believe we have demonstrated an effective automated inspection system for transmission lines by UAVs, while effectively integrating object detection techniques with UAV inspections, and our contributions are summarized below.

(1) To facilitate the deployment of the object detection model on the UAV, we chose Yolov5s as the baseline, and increased the value of mAP by 1.1 percentage points while ensuring that there was almost no decrease in speed, and the modified object detection model was able to detect grassland animals more accurately;

(2) To inspect transmission lines more efficiently and cost-effectively, we designed an automatic inspection system for transmission lines by UAVs. It includes path planning, intelligent aircraft nest, control algorithm and bird nest detection. After a large number of flight verification, this system greatly improves the inspection efficiency and reduces the labor cost. Meanwhile this system is integrated with object detection technology, which makes transmission line inspection more intelligent;

(3) A real-time inspection system based on YOLOX network model is designed to realize the high-precision automatic detection of high-voltage transmission lines by UAV. It includes a vision module, a control module and some strategies. The experiment proves that the system can effectively solve the problem that the target deviates from the center of the image when the UAV takes the image in the high-altitude inspection.

6.2 Future works

(1) The wildlife object detection model designed in Chapter 3 may not be applicable for the nighttime work due to the lack of nighttime dataset. We think that the problem of observing wildlife habits at night can be solved by hanging a searchlight on the UAV to collect photos of wildlife at night in the subsequent work.

(2) In terms of the UAV autonomous inspection system, at present we have only completed high-quality data collection for the linear towers in the electric tower types, and next we need to design reasonable path planning and correction strategies for other tower types.

(3) After completing the high-quality data collection, we plan to use the high-precision target detection model to detect various defects of the towers, such as, tilted insulator strings, displaced equalizing rings and missing locking pins. Intelligent inspection replaces traditional manual inspection, which will greatly reduce labor costs and improve inspection efficiency.

In the future, our fully autonomous electric power inspection robot is the trend to deepen its application in the power industry. Our team hopes that through continuous innovation and improvement works, we can better bring convenience to the grid overhead line inspection to become the real overhead line guardian, and jointly promote the popularization of intelligent life and the progress of the Internet of Things science.

Bibliography

[1] Jalil, B.; Leone, G. R.; Martinelli, M.; Moroni, D.; Pascali, M. A.; Berton, A. Fault detection in power equipment via an unmanned aerial system using multi modal data. *Sensors* **2019**, *19*, 3014.

[2] Menendez, O.; Cheein, F. A. A.; Perez, M.; Kouro, S. Robotics in power systems: Enabling a more reliable and safe grid. *IEEE Industrial Electronics Magazine* **2017**, *11*, 22–34.

[3] Disyadej, T.; Promjan, J.; Muneesawang, P.; Poochinapan, K.; Grzybowski, S. Application in O&M practices of overhead power line robotics. 2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia). 2019; pp 347–351.

[4] Wu, G.; Cao, H.; Xu, X.; Xiao, H.; Li, S.; Xu, Q.; Liu, B.; Wang, Q.; Wang, Z.; Ma, Y. Design and application of inspection system in a self-governing mobile robot system for high voltage transmission line inspection. 2009 Asia-Pacific Power and Energy Engineering Conference. 2009; pp 1–4.

[5] Jenssen, R.; Roverso, D.; others Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. *International Journal of Electrical Power & Energy Systems* **2018**, *99*, 107–120.

[6] https://www.eia.gov/todayinenergy/detail.php?id=48136 U.S. Energy InformationAdministration. (accessed on 1 May 2023).

[7] Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proceedings of the IEEE* **2023**,

[8] Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous detection and segmentation. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13. 2014; pp 297–312.

[9] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Proceedings of the IEEE international conference on computer vision. 2017; pp 2961–2969.

[10] Wu, Q.; Shen, C.; Wang, P.; Dick, A.; Van Den Hengel, A. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *40*, 1367–1381.

[11] Kang, K.; Li, H.; Yan, J.; Zeng, X.; Yang, B.; Xiao, T.; Zhang, C.; Wang, Z.; Wang, R.; Wang, X.; others T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology* **2017**, *28*, 2896–2907.

[12] LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.

[13] Viola, P.; Jones, M. 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. 2001; pp I–I.

[14] Viola, P.; Jones, M. J. Robust real-time face detection. *International journal of computer vision* **2004**, *57*, 137–154.

[15] Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). 2005; pp 886–893.

[16] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* **2004**, *60*, 91–110.

[17] Belongie, S.; Malik, J.; Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence* **2002**, *24*, 509–522.

[18] Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. 2008 IEEE conference on computer vision and pattern recognition. 2008; pp 1–8.

[19] Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D. Cascade object detection with deformable part models. 2010 IEEE Computer society conference on computer vision and pattern recognition. 2010; pp 2241–2248.

[20] Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence* **2009**, *32*, 1627–1645.

[21] Girshick, R. B. *From rigid templates to grammars: Object detection with structured models*; The University of Chicago, 2012.

[22] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM* **2017**, *60*, 84–90.

[23] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014; pp 580–587.

[24] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *38*, 142–158.

[25] Uijlings, J. R.; Van De Sande, K. E.; Gevers, T.; Smeulders, A. W. Selective search for object recognition. *International journal of computer vision* **2013**, *104*, 154–171.

[26] Girshick, R. B.; Felzenszwalb, P. F.; McAllester, D. Discriminatively trained deformable part models, release 5. **2012**,

[27] He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *37*, 1904–1916.

[28] Girshick, R. Fast r-cnn. Proceedings of the IEEE international conference on computer vision. 2015; pp 1440–1448.

[29] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **2015**, *28*.

[30] Zeiler, M. D.; Fergus, R. Visualizing and understanding convolutional networks. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13. 2014; pp 818–833.

[31] Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems* **2016**, *29*.

[32] Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Light-head r-cnn: In defense of two-stage object detector. *arXiv preprint arXiv:1711.07264* **2017**,

[33] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016; pp 779–788.

[34] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* **2018**,

[35] Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017; pp 7263–7271.

[36] Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* **2020**,

[37] Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696* **2022**,

[38] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. Ssd: Single shot multibox detector. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. 2016; pp 21–37.

[39] Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv preprint arXiv:1904.07850* **2019**,

[40] Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. Proceedings of the European conference on computer vision (ECCV). 2018; pp 734–750.

[41] Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019; pp 850–859.

[42] Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. 2020; pp 213–229.

[43] Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* **2020**,

[44] Malisiewicz, T.; Gupta, A.; Efros, A. A. Ensemble of exemplar-svms for object detection and beyond. 2011 International conference on computer vision. 2011; pp 89–96.

[45] Malisiewicz, T. *Exemplar-based representations for object detection, association and beyond*; Carnegie Mellon University, 2011.

[46] Hosang, J.; Benenson, R.; Dollár, P.; Schiele, B. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence* **2015**, *38*, 814–830.

[47] Hosang, J.; Benenson, R.; Schiele, B. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962* **2014**,

[48] Alexe, B.; Deselaers, T.; Ferrari, V. What is an object? 2010 IEEE computer society conference on computer vision and pattern recognition. 2010; pp 73–80.

[49] Alexe, B.; Deselaers, T.; Ferrari, V. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence* **2012**, *34*, 2189–2202.

[50] Cheng, M.-M.; Zhang, Z.; Lin, W.-Y.; Torr, P. BING: Binarized normed gradients for objectness estimation at 300fps. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014; pp 3286–3293.

[51] Erhan, D.; Szegedy, C.; Toshev, A.; Anguelov, D. Scalable object detection using deep neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014; pp 2147–2154.

[52] Szegedy, C.; Toshev, A.; Erhan, D. Deep neural networks for object detection. *Advances in neural information processing systems* **2013**, *26*.

[53] Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. Proceedings of the IEEE/CVF international conference on computer vision. 2019; pp 9657–9666.

[54] Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. Proceedings of the IEEE/CVF international conference on computer vision. 2019; pp 6569–6578.

[55] Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. Proceedings of the IEEE/CVF international conference on computer vision. 2019; pp 9627–9636.

[56] Cai, Z.; Fan, Q.; Feris, R. S.; Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. 2016; pp 354–370.

[57] Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018; pp 6154–6162.

[58] Divvala, S. K.; Hoiem, D.; Hays, J. H.; Efros, A. A.; Hebert, M. An empirical study of context in object detection. 2009 IEEE Conference on computer vision and Pattern Recognition. 2009; pp 1271–1278.

[59] Torralba, A.; Sinha, P. *Detecting faces in impoverished images*; 2001.

[60] Zagoruyko, S.; Lerer, A.; Lin, T.-Y.; Pinheiro, P. O.; Gross, S.; Chintala, S.; Dollár, P. A multipath network for object detection. *arXiv preprint arXiv:1604.02135* **2016**,

[61] Zeng, X.; Ouyang, W.; Yang, B.; Yan, J.; Wang, X. Gated bi-directional cnn for object detection. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. 2016; pp 354–369.

[62] Zeng, X.; Ouyang, W.; Yan, J.; Li, H.; Xiao, T.; Wang, K.; Liu, Y.; Zhou, Y.; Yang, B.; Wang, Z.; others Crafting gbd-net for object detection. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *40*, 2109–2123.

[63] Ouyang, W.; Wang, K.; Zhu, X.; Wang, X. Learning chained deep features and classifiers for cascade in object detection. *arXiv preprint arXiv:1702.07054* **2017**,

[64] Zhu, Y.; Zhao, C.; Wang, J.; Zhao, X.; Wu, Y.; Lu, H. Couplenet: Coupling global structure with local parts for object detection. Proceedings of the IEEE international conference on computer vision. 2017; pp 4126–4134.

[65] Li, Z.; Chen, Y.; Yu, G.; Deng, Y. R-fcn++: Towards accurate region-based fully convolutional networks for object detection. Proceedings of the AAAI conference on artificial intelligence. 2018.

[66] Liu, S.; Huang, D.; others Receptive field block net for accurate and fast object detection. Proceedings of the European conference on computer vision (ECCV). 2018; pp 385–400.

[67] Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018; pp 7794–7803.

[68] Bell, S.; Zitnick, C. L.; Bala, K.; Girshick, R. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016; pp 2874–2883.

[69] Li, J.; Wei, Y.; Liang, X.; Dong, J.; Xu, T.; Feng, J.; Yan, S. Attentive contexts for object detection. *IEEE Transactions on Multimedia* **2016**, *19*, 944–954.

[70] Song, Z.; Chen, Q.; Huang, Z.; Hua, Y.; Yan, S. Contextualizing object detection and classification. CVPR 2011. 2011; pp 1585–1592.

[71] Chen, X.; Gupta, A. Spatial memory for context reasoning in object detection. Proceedings of the IEEE international conference on computer vision. 2017; pp 4086–4096.

[72] Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018; pp 3588–3597.

[73] Pato, L. V.; Negrinho, R.; Aguiar, P. M. Seeing without looking: Contextual rescoring of object detections for ap maximization. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020; pp 14610–14618.

[74] Gupta, S.; Hariharan, B.; Malik, J. Exploring person context and local scene context for object detection. *arXiv preprint arXiv:1511.08177* **2015**,

[75] Liu, Y.; Wang, R.; Shan, S.; Chen, X. Structure inference net: Object detection using scene-level context and instance-level relationships. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018; pp 6985–6994.

[76] Papageorgiou, C.; Poggio, T. A trainable system for object detection. *International journal of computer vision* **2000**, *38*, 15–33.

[77] Rothe, R.; Guillaumin, M.; Van Gool, L. Non-maximum suppression for object detection by passing messages between windows. Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part I 12. 2015; pp 290–306.

[78] Bodla, N.; Singh, B.; Chellappa, R.; Davis, L. S. Soft-NMS–improving object detection with one line of code. Proceedings of the IEEE international conference on computer vision. 2017; pp 5561–5569.

[79] He, Y.; Zhu, C.; Wang, J.; Savvides, M.; Zhang, X. Bounding box regression with uncertainty for accurate object detection. Proceedings of the ieee/cvf conference on computer vision and pattern recognition. 2019; pp 2888–2897.

[80] Liu, S.; Huang, D.; Wang, Y. Adaptive nms: Refining pedestrian detection in a crowd. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019; pp 6459–6468.

[81] Mrowca, D.; Rohrbach, M.; Hoffman, J.; Hu, R.; Saenko, K.; Darrell, T. Spatial semantic regularisation for large scale object detection. Proceedings of the IEEE international conference on computer vision. 2015; pp 2003–2011.

[82] Solovyev, R.; Wang, W.; Gabruseva, T. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing* **2021**, *107*, 104117.

[83] Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics* **2021**, *52*, 8574–8586.

[84] Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* **2013**,

[85] Vaillant, R.; Monrocq, C.; Le Cun, Y. Original approach for the localisation of objects in images. *IEE Proceedings-Vision, Image and Signal Processing* **1994**, *141*, 245–250.

[86] Hosang, J.; Benenson, R.; Schiele, B. Learning non-maximum suppression. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017; pp 4507–4515.

[87] Tan, Z.; Nie, X.; Qian, Q.; Li, N.; Li, H. Learning to rank proposals for object detection. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019; pp 8273–8281.

[88] Wang, J.; Song, L.; Li, Z.; Sun, H.; Sun, J.; Zheng, N. End-to-end object detection with fully convolutional network. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021; pp 15849–15858.

[89] Luo, Y.; Yu, X.; Yang, D.; Zhou, B. A survey of intelligent transmission line inspection based on unmanned aerial vehicle. *Artificial Intelligence Review* **2023**, *56*, 173–201.

[90] Luo, Y.; Zhang, X.; Yang, D.; Sun, Q. Emission trading based optimal scheduling strategy of energy hub with energy storage and integrated electric vehicles. *Journal of Modern Power Systems and Clean Energy* **2020**, *8*, 267–275.

[91] He, T.; Zeng, Y.; Hu, Z. Research of multi-rotor UAVs detailed autonomous inspection technology of transmission lines based on route planning. *IEEE Access* **2019**, *7*, 114955–114965.

[92] Valianti, P.; Papaioannou, S.; Kolios, P.; Ellinas, G. Multi-agent coordinated close-in jamming for disabling a rogue drone. *IEEE Transactions on Mobile Computing* **2021**, *21*, 3700–3717.

[93] Fotohi, R. Securing of Unmanned Aerial Systems (UAS) against security threats using human immune system. *Reliability Engineering & System Safety* **2020**, *193*, 106675.

[94] Park, J.-Y.; Lee, J.-K.; Cho, B.-H.; Oh, K.-Y. An inspection robot for live-line suspension insulator strings in 345-kV power lines. *IEEE Transactions on power delivery* **2012**, *27*, 632–639.

[95] Song, L.; Wang, H.; Chen, P. Automatic patrol and inspection method for machinery diagnosis robot—Sound signal-based fuzzy search approach. *IEEE Sensors Journal* **2020**, *20*, 8276–8286.

[96] Guerreiro, B. J.; Silvestre, C.; Cunha, R.; Cabecinhas, D. LiDAR-based control of autonomous rotorcraft for the inspection of pierlike structures. *IEEE Transactions on Control Systems Technology* **2017**, *26*, 1430–1438.

[97] Morrell, B.; Thakker, R.; Merewether, G.; Reid, R.; Rigter, M.; Tzanetos, T.; Chamitoff, G. Comparison of trajectory optimization algorithms for high-speed quadrotor flight near obstacles. *IEEE Robotics and Automation Letters* **2018**, *3*, 4399–4406.

[98] Wang, H.; Yang, G.; Li, E.; Tian, Y.; Zhao, M.; Liang, Z. High-voltage power transmission tower detection based on faster R-CNN and YOLO-V3. 2019 Chinese Control Conference (CCC). 2019; pp 8750–8755.

[99] Ferdaus, M. M.; Anavatti, S. G.; Pratama, M.; Garratt, M. A. Towards the use of fuzzy logic systems in rotary wing unmanned aerial vehicle: a review. *Artificial Intelligence Review* **2020**, *53*, 257–290.

[100] Dong, F.; You, K.; Zhang, J. Flight control for UAV loitering over a ground target with unknown maneuver. *IEEE Transactions on Control Systems Technology* **2019**, *28*, 2461–2473.

[101] Bauersfeld, L.; Spannagl, L.; Ducard, G. J.; Onder, C. H. MPC flight control for a tilt-rotor VTOL aircraft. *IEEE Transactions on Aerospace and Electronic Systems* **2021**, *57*, 2395–2409.

[102] Oubbati, O. S.; Lakas, A.; Lorenz, P.; Atiquzzaman, M.; Jamalipour, A. Leveraging communicating UAVs for emergency vehicle guidance in urban areas. *IEEE Transactions on Emerging Topics in Computing* **2019**, *9*, 1070–1082.

[103] Yang, S.; Xian, B. Energy-based nonlinear adaptive control design for the quadrotor UAV system with a suspended payload. *IEEE Transactions on Industrial Electronics* **2019**, *67*, 2054–2064.

[104] Labbadi, M.; Cherkaoui, M. Adaptive fractional-order nonsingular fast terminal sliding mode based robust tracking control of quadrotor UAV with Gaussian random disturbances and uncertainties. *IEEE Transactions on Aerospace and Electronic Systems* **2021**, *57*, 2265–2277.

[105] Mejias, L.; Campoy, P.; Saripalli, S.; Sukhatme, G. S. A visual servoing approach for tracking features in urban areas using an autonomous helicopter. Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. 2006; pp 2503–2508.

[106] Montambault, S.; Beaudry, J.; Toussaint, K.; Pouliot, N. On the application of VTOL UAVs to the inspection of power utility assets. 2010 1st international conference on applied robotics for the power industry. 2010; pp 1–7.

[107] Pouliot, N.; Richard, P.-L.; Montambault, S. LineScout technology opens the way to robotic inspection and maintenance of high-voltage power lines. *IEEE Power and Energy Technology Systems Journal* **2015**, *2*, 1–11.

[108] Zhang, Y.; Yuan, X.; Li, W.; Chen, S. Automatic power line inspection using UAV images. *Remote Sensing* **2017**, *9*, 824.

[109] Tang, M.-w.; Dai, L.-h.; Lin, C.-h.; Wang, F.-d.; Song, F.-g. Application of unmanned aerial vehicle in inspecting transmission lines. *Electric Power* **2013**, *46*, 35–38.

[110] Yadav, M.; Chousalkar, C. G. Extraction of power lines using mobile LiDAR data of roadway environment. *Remote Sensing Applications: Society and Environment* **2017**, *8*, 258–265.

[111] Negassi, M.; Suarez-Ibarrola, R.; Hein, S.; Miernik, A.; Reiterer, A. Application of artificial neural networks for automated analysis of cystoscopic images: a review of the current status and future prospects. *World journal of urology* **2020**, *38*, 2349–2358.

[112] Wang, S.-J.; Kuo, L.-C.; Jong, H.-H.; Wu, Z.-H. Representing images using points on image surfaces. *IEEE transactions on image processing* **2005**, *14*, 1043–1056.

[113] Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; others Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* **2012**, *29*, 82–97.

[114] Salloum, S.; Huang, J. Z.; He, Y. Random sample partition: a distributed data model for big data analysis. *IEEE Transactions on Industrial Informatics* **2019**, *15*, 5846–5854.

[115] Ghesu, F. C.; Krubasik, E.; Georgescu, B.; Singh, V.; Zheng, Y.; Hornegger, J.; Comaniciu, D. Marginal space deep learning: efficient architecture for volumetric image parsing. *IEEE transactions on medical imaging* **2016**, *35*, 1217–1228.

[116] She, L.; Fan, Y.; Wang, J.; Cai, L.; Xue, J.; Xu, M. Insulator surface breakage recognition based on multiscale residual neural network. *IEEE Transactions on Instrumentation and Measurement* **2021**, *70*, 1–9.

[117] Peng, X.; Yang, F.; Wang, G.; Wu, Y.; Li, L.; Li, Z.; Bhatti, A. A.; Zhou, C.; Hepburn, D. M.; Reid, A. J.; others A convolutional neural network-based deep learning methodology for recognition of partial discharge patterns from high-voltage cables. *IEEE Transactions on Power Delivery* **2019**, *34*, 1460–1469.

[118] Guo, Y.; Wu, M.; Tang, K.; Tie, J.; Li, X. Covert spoofing algorithm of UAV based on GPS/INS-integrated navigation. *IEEE Transactions on Vehicular Technology* **2019**, *68*, 6557–6564.

[119] Kim, Y.; An, J.; Lee, J. Robust navigational system for a transporter using GPS/INS fusion. *IEEE Transactions on Industrial Electronics* **2017**, *65*, 3346–3354.

[120] Duan, H.; Xin, L.; Shi, Y. Homing pigeon-inspired autonomous navigation system for unmanned aerial vehicles. *IEEE Transactions On Aerospace and Electronic Systems* **2021**, *57*, 2218–2224.

[121] Chen, P.; Dang, Y.; Liang, R.; Zhu, W.; He, X. Real-time object tracking on a drone with multi-inertial sensing data. *IEEE Transactions on Intelligent Transportation Systems* **2017**, *19*, 131–139.

[122] Accardo, D.; Fasano, G.; Forlenza, L.; Moccia, A.; Rispoli, A. Flight test of a radar-based tracking system for UAS sense and avoid. *IEEE Transactions on Aerospace and Electronic Systems* **2013**, *49*, 1139–1160.

[123] Faraji-Biregani, M.; Fotohi, R. Secure communication between UAVs using a method based on smart agents in unmanned aerial vehicles. *The journal of supercomputing* **2021**, *77*, 5076–5103.

[124] Yong, H.; Huang, J.; Xiang, W.; Hua, X.; Zhang, L. Panoramic background image generation for PTZ cameras. *IEEE Transactions on Image Processing* **2019**, *28*, 3162–3176.

[125] Varcheie, P. D. Z.; Bilodeau, G.-A. Adaptive fuzzy particle filter tracker for a PTZ camera in an IP surveillance system. *IEEE Transactions on instrumentation and measurement* **2010**, *60*, 354–371.

[126] Liu, Y.; Chen, Y.; Jiao, Y.; Ma, H.; Wu, T. A shared satellite ground station using user-oriented virtualization technology. *IEEE Access* **2020**, *8*, 63923–63934.

[127] Rucco, A.; Sujit, P.; Aguiar, A. P.; De Sousa, J. B.; Pereira, F. L. Optimal rendezvous trajectory for unmanned aerial-ground vehicles. *IEEE Transactions on Aerospace and Electronic Systems* **2017**, *54*, 834–847.

[128] Li, Z.; Namiki, A.; Suzuki, S.; Wang, Q.; Zhang, T.; Wang, W. Application of low-altitude UAV remote sensing image object detection based on improved YOLOv5. *Applied Sciences* **2022**, *12*, 8314.

[129] Choiński, M.; Rogowski, M.; Tynecki, P.; Kuijper, D. P.; Churski, M.; Bubnicki, J. W. A first step towards automated species recognition from camera trap images of mammals using AI in a European temperate forest. Computer Information Systems and Industrial Management: 20th International Conference, CISIM 2021, Ełk, Poland, September 24–26, 2021, Proceedings 20. 2021; pp 299–310.

[130] Lema, D. G.; Pedrayes, O. D.; Usamentiaga, R.; García, D. F.; Alonso, Á. Cost-performance evaluation of a recognition service of livestock activity using aerial images. *Remote Sensing* **2021**, *13*, 2318.

[131] Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A forest fire detection system based on ensemble learning. *Forests* **2021**, *12*, 217.

[132] Rahman, E. U.; Zhang, Y.; Ahmad, S.; Ahmad, H. I.; Jobaer, S. Autonomous vision-based primary distribution systems porcelain insulators inspection using UAVs. *Sensors* **2021**, *21*, 974.

[133] Li, X.; Li, X.; Pan, H. Multi-scale vehicle detection in high-resolution aerial images with context information. *IEEE Access* **2020**, *8*, 208643–208657.

[134] Han, X.; Chang, J.; Wang, K. Real-time object detection based on YOLO-v2 for tiny vehicle object. *Procedia Computer Science* **2021**, *183*, 61–72.

[135] Adami, D.; Ojo, M. O.; Giordano, S. Design, Development and Evaluation of an Intelligent Animal Repelling System for Crop Protection Based on Embedded Edge-AI. *IEEE Access* **2021**, *9*, 132125–132139.

[136] Chen, L.; Zheng, M.; Duan, S.; Luo, W.; Yao, L. Underwater target recognition based on improved YOLOv4 neural network. *Electronics* **2021**, *10*, 1634.

[137] Peng, J.; Wang, D.; Liao, X.; Shao, Q.; Sun, Z.; Yue, H.; Ye, H. Wild animal survey using UAS imagery and deep learning: modified Faster R-CNN for kiang detection in Tibetan Plateau. *ISPRS Journal of Photogrammetry and Remote Sensing* **2020**, *169*, 364–376.

[138] Yan, L.; Miao, Z.; Zhang, W. Pig face detection method based on improved CenterNet algorithm. 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI). 2022; pp 174–179.

[139] Xu, X.; Zhang, X.; Yu, B.; Hu, X. S.; Rowen, C.; Hu, J.; Shi, Y. Dac-sdc low power object detection challenge for uav applications. *IEEE transactions on pattern analysis and machine intelligence* **2019**, *43*, 392–403.

[140] Yu, W.; Yang, T.; Chen, C. Towards resolving the challenge of long-tail distribution in UAV images for object detection. Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2021; pp 3258–3267.

[141] Zhang, R.; Shao, Z.; Huang, X.; Wang, J.; Li, D. Object detection in UAV images via global density fused convolutional network. *Remote Sensing* **2020**, *12*, 3140.

[142] Zhang, P.; Zhong, Y.; Li, X. SlimYOLOv3: Narrower, faster and better for real-time UAV applications. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019; pp 0–0.

[143] Hu, Y.; Wu, X.; Zheng, G.; Liu, X. Object detection of UAV for anti-UAV based on improved YOLO v3. 2019 Chinese Control Conference (CCC). 2019; pp 8386–8390.

[144] Liu, M.; Wang, X.; Zhou, A.; Fu, X.; Ma, Y.; Piao, C. Uav-yolo: Small object detection on unmanned aerial vehicle perspective. *Sensors* **2020**, *20*, 2238.

[145] Zhang, H.; Sun, M.; Li, Q.; Liu, L.; Liu, M.; Ji, Y. An empirical study of multi-scale object detection in high resolution UAV images. *Neurocomputing* **2021**, *421*, 173–182.

[146] Tirandaz, Z.; Akbarizadeh, G. A two-phase algorithm based on kurtosis curvelet energy and unsupervised spectral regression for segmentation of SAR images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2015**, *9*, 1244–1264.

[147] Zalpour, M.; Akbarizadeh, G.; Alaei-Sheini, N. A new approach for oil tank detection using deep learning features with control false alarm rate in high-resolution satellite imagery. *International Journal of Remote Sensing* **2020**, *41*, 2239–2262.

[148] Jocher, G. https://doi.org/10.5281/zenodo.6222936. *(accessed on 5 June 2022).*

[149] Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017; pp 2117–2125.

[150] Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019; pp 658–666.

[151] Hu, D.; Zhang, Y.; Xufeng, L.; Zhang, X. Detection of material on a tray in automatic assembly line based on convolutional neural network. *IET Image Processing* **2021**, *15*, 3400–3409.

[152] Afzaal, H.; Farooque, A. A.; Schumann, A. W.; Hussain, N.; McKenzie-Gopsill, A.; Esau, T.; Abbas, F.; Acharya, B. Detection of a potato disease (early blight) using artificial intelligence. *Remote Sensing* **2021**, *13*, 411.

[153] Biffi, L. J.; Mitishita, E.; Liesenberg, V.; Santos, A. A. d.; Gonçalves, D. N.; Estrabis, N. V.; Silva, J. d. A.; Osco, L. P.; Ramos, A. P. M.; Centeno, J. A. S.; others ATSS deep learning-based approach to detect apple fruits. *Remote Sensing* **2020**, *13*, 54.

[154] Yan, B.; Fan, P.; Lei, X.; Liu, Z.; Yang, F. A real-time apple targets detection method for picking robot based on improved YOLOv5. *Remote Sensing* **2021**, *13*, 1619.

[155] Fu, L.; Gao, F.; Wu, J.; Li, R.; Karkee, M.; Zhang, Q. Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. *Computers and Electronics in Agriculture* **2020**, *177*, 105687.

[156] Messinis, S.; Vosniakos, G. C. An agent-based flexible manufacturing system controller with Petri-net enabled algebraic deadlock avoidance. *Reports in Mechanical Engineering* **2020**, *1*, 77–92.

[157] Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018; pp 8759–8768.

[158] Bai, M.; Urtasun, R. Deep watershed transform for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017; pp 5221–5229.

[159] Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016; pp 3213–3223.

[160] Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; others Speed/accuracy trade-offs for modern convolutional object detectors. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017; pp 7310–7311.

[161] Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large kernel matters–improve semantic segmentation by global convolutional network. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017; pp 4353–4361.

[162] Mirzazadeh, A.; Azizi, A.; Abbaspour-Gilandeh, Y.; Hernández-Hernández, J. L.; Hernández-Hernández, M.; Gallardo-Bernal, I. A novel technique for classifying bird damage to rapeseed plants based on a deep learning algorithm. *Agronomy* **2021**, *11*, 2364.

[163] Newell, A.; Yang, K.; Deng, J. Stacked hourglass networks for human pose estimation. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. 2016; pp 483–499.

[164] Elsken, T.; Metzen, J. H.; Hutter, F. Efficient multi-objective neural architecture search via lamarckian evolution. *arXiv preprint arXiv:1804.09081* **2018**,

[165] Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* **2018**,

[166] Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018; pp 7132–7141.

[167] GitHub https://github.com/sczhengyabin/Image-Downloader. *(accessed on 30 June 2022).*

[168] Group, V. C. https://www.vcg.com/creative-video. *(accessed on 30 June 2022).*

[169] Azizi, A.; Abbaspour-Gilandeh, Y.; Mesri-Gundoshmian, T.; Farooque, A. A.; Afzaal, H. Estimation of soil surface roughness using stereo vision approach. *Sensors* **2021**, *21*, 4386.

[170] YouTube https://www.youtube.com/watch?v=FVtpfy10AJM. *(accessed on 30 June 2022.*

[171] Li, Z.; Zhang, Y.; Wu, H.; Suzuki, S.; Namiki, A.; Wang, W. Design and Application of a UAV Autonomous Inspection System for High-Voltage Power Transmission Lines. *Remote Sensing* **2023**, *15*, 865.

[172] Mao, T.; Huang, K.; Zeng, X.; Ren, L.; Wang, C.; Li, S.; Zhang, M.; Chen, Y. Development of power transmission line defects diagnosis system for UAV inspection based on binocular depth imaging technology. 2019 2nd International Conference on Electrical Materials and Power Equipment (ICEMPE). 2019; pp 478–481.

[173] Wu, C.; Song, J.-g.; Zhou, H.; Yang, X.-f.; Ni, H.-y.; Yan, W.-x. Research on intelligent inspection system for HV power transmission lines. 2020 IEEE International Conference on High Voltage Engineering and Application (ICHVE). 2020; pp 1–4.

[174] Knapik, W.; Kowalska, M. K.; Odlanicka-Poczobutt, M.; Kasperek, M. The Internet of Things through Internet Access Using an Electrical Power Transmission System (Power Line Communication) to Improve Digital Competencies and Quality of Life of Selected Social Groups in Poland's Rural Areas. *Energies* **2022**, *15*, 5018.

[175] Zhang, Y.; Yuan, X.; Fang, Y.; Chen, S. UAV low altitude photogrammetry for power line inspection. *ISPRS International Journal of GEO-information* **2017**, *6*, 14.

[176] Chen, D.-Q.; Guo, X.-H.; Huang, P.; Li, F.-H. Safety distance analysis of 500kv transmission line tower uav patrol inspection. *IEEE Letters on Electromagnetic Compatibility Practice and Applications* **2020**, *2*, 124–128.

[177] Larrauri, J. I.; Sorrosal, G.; González, M. Automatic system for overhead power line inspection using an Unmanned Aerial Vehicle—RELIFO project. 2013 International conference on unmanned aircraft systems (ICUAS). 2013; pp 244–252.

[178] Vemula, S.; Frye, M. Mask R-CNN Powerline Detector: A Deep Learning approach with applications to a UAV. 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC). 2020; pp 1–6.

[179] Zhao, Z.; Qi, H.; Qi, Y.; Zhang, K.; Zhai, Y.; Zhao, W. Detection method based on automatic visual shape clustering for pin-missing defect in transmission lines. *IEEE Transactions on Instrumentation and Measurement* **2020**, *69*, 6080–6091.

[180] Debenest, P.; Guarnieri, M.; Takita, K.; Fukushima, E. F.; Hirose, S.; Tamura, K.; Kimura, A.; Kubokawa, H.; Iwama, N.; Shiga, F. Expliner-Robot for inspection of transmission lines. 2008 IEEE International Conference on Robotics and Automation. 2008; pp 3978–3984.

[181] Matikainen, L.; Lehtomäki, M.; Ahokas, E.; Hyyppä, J.; Karjalainen, M.; Jaakkola, A.; Kukko, A.; Heinonen, T. Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote sensing* **2016**, *119*, 10–31.

[182] Finotto, V.; Horikawa, O.; Hirakawa, A.; Chamas Filho, A. Pole type robot for distribution power line inspection. 2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI). 2012; pp 88–93.

[183] Martinez, C.; Sampedro, C.; Chauhan, A.; Collumeau, J. F.; Campoy, P. The Power Line Inspection Software (PoLIS): A versatile system for automating power line inspection. *Engineering applications of artificial intelligence* **2018**, *71*, 293–314.

[184] Li, J.; Wang, L.; Shen, X. Unmanned aerial vehicle intelligent patrol-inspection system applied to transmission grid. 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2). 2018; pp 1–5.

[185] Calvo, A.; Silano, G.; Capitán, J. Mission planning and execution in heterogeneous teams of aerial robots supporting power line inspection operations. 2022 International Conference on Unmanned Aircraft Systems (ICUAS). 2022; pp 1644–1649.

[186] Luque-Vega, L. F.; Castillo-Toledo, B.; Loukianov, A.; Gonzalez-Jimenez, L. E. Power line inspection via an unmanned aerial system based on the quadrotor helicopter. MELECON 2014-2014 17th IEEE Mediterranean electrotechnical conference. 2014; pp 393–397.

[187] Li, Z.; Mu, S.; Li, J.; Wang, W.; Liu, Y. Transmission line intelligent inspection central control and mass data processing system and application based on UAV. 2016 4th International Conference on Applied Robotics for the Power Industry (CARPI). 2016; pp 1–5.

[188] Guan, H.; Sun, X.; Su, Y.; Hu, T.; Wang, H.; Wang, H.; Peng, C.; Guo, Q. UAV-lidar aids automatic intelligent powerline inspection. *International Journal of Electrical Power & Energy Systems* **2021**, *130*, 106987.

[189] Xu, C.; Li, Q.; Zhou, Q.; Zhang, S.; Yu, D.; Ma, Y. Power line-guided automatic electric transmission line inspection system. *IEEE Transactions on Instrumentation and Measurement* **2022**, *71*, 1–18.

[190] Li, H.; Dong, Y.; Liu, Y.; Ai, J. Design and Implementation of UAVs for Bird's Nest Inspection on Transmission Lines Based on Deep Learning. *Drones* **2022**, *6*, 252.

[191] Hao, J.; Wulin, H.; Jing, C.; Xinyu, L.; Xiren, M.; Shengbin, Z. Detection of bird nests on power line patrol using single shot detector. 2019 Chinese Automation Congress (CAC). 2019; pp 3409–3414.

[192] Jenssen, R.; Roverso, D.; others Intelligent monitoring and inspection of power line components powered by UAVs and deep learning. *IEEE Power and energy technology systems journal* **2019**, *6*, 11–21.

[193] Yang, L.; Fan, J.; Song, S.; Liu, Y. A light defect detection algorithm of power insulators from aerial images for power inspection. *Neural Computing and Applications* **2022**, *34*, 17951–17961.

[194] Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430* **2021**,

[195] Wang, W.; Ma, H.; Xia, M.; Weng, L.; Ye, X. Attitude and altitude controller design for quad-rotor type MAVs. *Mathematical Problems in Engineering* **2013**, *2013*.

[196] Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision. 2017; pp 2980–2988.

[197] Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021; pp 13713–13722.

[198] Zhang, H.; Wang, Y.; Dayoub, F.; Sunderhauf, N. Varifocalnet: An iou-aware dense object detector. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021; pp 8514–8523.

[199] Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv preprint arXiv:2205.12740* **2022**,

[200] Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv preprint arXiv:1911.09516* **2019**,

[201] Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I. S. Cbam: Convolutional block attention module. Proceedings of the European conference on computer vision (ECCV). 2018; pp 3–19.

[202] Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020; pp 11534–11542.

[203] Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. Proceedings of the AAAI conference on artificial intelligence. 2020; pp 12993–13000.

[204] Chunxiang, Z.; Jiacheng, Q.; Wang, B. YOLOX on Embedded Device With CCTV & TensorRT for Intelligent Multicategories Garbage Identification and Classification. *IEEE Sensors Journal* **2022**, *22*, 16522–16532.

[205] Rao, Y.; Zhao, W.; Tang, Y.; Zhou, J.; Lim, S. N.; Lu, J. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems* **2022**, *35*, 10353–10366.

[206] Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; Ren, J. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems* **2022**, *35*, 12934–12949.

[207] Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021; pp 13733–13742.

[208] Li, Y.; Wu, C.-Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. Mvitv2: Improved multiscale vision transformers for classification and detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022; pp 4804–4814.

[209] Li, Z.; Wang, Q.; Zhang, T.; Ju, C.; Suzuki, S.; Namiki, A. UAV High-voltage Power Transmission Line Autonomous Correction Inspection System Based on Object Detection. *IEEE Sensors Journal* **2023**,

[210] Vom Bögel, G.; Cousin, L.; Iversen, N.; Ebeid, E. S. M.; Hennig, A. Drones for inspection of overhead power lines with recharge function. 2020 23rd Euromicro Conference on Digital System Design (DSD). 2020; pp 497–502.

[211] Togola, S.; Kiemde, S. M. A.; Kora, A. D. Real Time and Post-Processing Flight Inspection by Drone: A Survey. 2020 43rd International Conference on Telecommunications and Signal Processing (TSP). 2020; pp 399–402.

[212] Iversen, N.; Schofield, O. B.; Cousin, L.; Ayoub, N.; Vom Bögel, G.; Ebeid, E. Design, integration and implementation of an intelligent and self-recharging drone system for autonomous power line inspection. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2021; pp 4168–4175.

[213] Shuangchun, S.; Yanlei, L.; Zhenxiao, Y.; Kai, W.; Ping, Y.; Kun, Z.; Tao, Y. Review of autonomous inspection technology for power lines using uavs. 2021 IEEE International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT). 2021; pp 481–484.

[214] Ma, Y.; Li, Q.; Chu, L.; Zhou, Y.; Xu, C. Real-time detection and spatial localization of insulators for UAV inspection based on binocular stereo vision. *Remote Sensing* **2021**, *13*, 230.

[215] Huang, L.; Yang, Y.; Deng, Y.; Yu, Y. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874* **2015**,

[216] Guo, Q.; Liu, J.; Kaliuzhnyi, M. YOLOX-SAR: High-Precision Object Detection System Based on Visible and Infrared Sensors for SAR Remote Sensing. *IEEE Sensors Journal* **2022**, *22*, 17243–17253.

[217] Jia, W.; Xu, S.; Liang, Z.; Zhao, Y.; Min, H.; Li, S.; Yu, Y. Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. *IET Image Processing* **2021**, *15*, 3623–3637.

[218] Liu, C.; Wu, Y.; Liu, J.; Sun, Z. Improved YOLOv3 network for insulator detection in aerial images with diverse background interference. *Electronics* **2021**, *10*, 771.

[219] Li, Q.; Zhao, F.; Xu, Z.; Wang, J.; Liu, K.; Qin, L. Insulator and damage detection and location based on YOLOv5. 2022 International Conference on Power Energy Systems and Applications (ICoPESA). 2022; pp 17–24.

[220] Liu, X.; Jiang, H.; Chen, J.; Chen, J.; Zhuang, S.; Miao, X. Insulator detection in aerial images based on faster regions with convolutional neural network. 2018 IEEE 14th international conference on control and automation (ICCA). 2018; pp 1082–1086.

[221] Miao, X.; Liu, X.; Chen, J.; Zhuang, S.; Fan, J.; Jiang, H. Insulator detection in aerial images for transmission line inspection using single shot multibox detector. *IEEE Access* **2019**, *7*, 9945–9956.

[222] Wang, Z.; Liu, X.; Peng, H.; Zheng, L.; Gao, J.; Bao, Y. Railway insulator detection based on adaptive cascaded convolutional neural network. *IEEE Access* **2021**, *9*, 115676–115686.

[223] Hui, X.; Bian, J.; Yu, Y.; Zhao, X.; Tan, M. A novel autonomous navigation approach for UAV power line inspection. 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). 2017; pp 634–639.

[224] Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; others Searching for mobilenetv3. Proceedings of the IEEE/CVF international conference on computer vision. 2019; pp 1314–1324.

[225] Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020; pp 1580–1589.

[226] He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X.-S. Alpha-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression. *Advances in Neural Information Processing Systems* **2021**, *34*, 20230–20242.

[227] Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems* **2021**,

[228] Nota, E.; Nijland, W.; de Haas, T. Improving UAV-SfM time-series accuracy by co-alignment and contributions of ground control or RTK positioning. *International Journal of Applied Earth Observation and Geoinformation* **2022**, *109*, 102772.

List of Contributions

Journal papers:

(1) Z. Li, A. Namiki, S. Suzuki, Q. Wang, T. Zhang, and W. Wang, "Application of Low-Altitude UAV Remote Sensing Image Object Detection Based on Improved YOLOv5," *Appl. Sci.*, vol. 12, no. 16, p. 8314, Aug. 2022.

(2) Z. Li, Y. Zhang, H. Wu, S. Suzuki, A. Namiki, and W. Wang, "Design and Application of a UAV Autonomous Inspection System for High-Voltage Power Transmission Lines," *Remote Sens.*, vol. 15, no. 3, p. 865, Feb. 2023.

(3) Z. Li, Q. Wang, T. Zhang, C. Ju, S. Suzuki and A. Namiki, "UAV High-voltage Power Transmission Line Autonomous Correction Inspection System Based on Object Detection," *IEEE Sens. J.*, vol. 23, no. 9, pp. 10215-10230, 1 May1, 2023, doi: 10.1109/JSEN.2023.3260360.

Conference papers:

(1) Ziran Li, Akio Namiki. UAV Position Correction System Based on YOLOX. "The Robotics and Mechatronics Conference 2023", presentation in Nagoya. Jun. 2023.