

Article

Design and Analysis of Refined Inspection of Field Conditions of Oilfield Pumping Wells Based on Rotorcraft UAV Technology

Yu Zhou ¹, Chunxue Wu ¹ , Qunhui Wu ², Zelda Makati Eli ¹, Naixue Xiong ¹  and Sheng Zhang ^{1,*}

¹ School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; zhouyu0509@126.com (Y.Z.); wcx@usst.edu.cn (C.W.); m18097909971@163.com (Z.M.E.); xionгнаixue@gmail.com (N.X.)

² Shanghai HEST Co. Ltd., Shanghai 201610, China; shhest@aliyun.com

* Correspondence: zhangsheng_usst@aliyun.com; Tel.: +86-1338-6002-013

Received: 13 October 2019; Accepted: 28 November 2019; Published: 9 December 2019



Abstract: The traditional oil well monitoring method relies on manual acquisition and various high-precision sensors. Using the indicator diagram to judge the working condition of the well is not only difficult to establish but also consumes huge manpower and financial resources. This paper proposes the use of computer vision in the detection of working conditions in oil extraction. Combined with the advantages of an unmanned aerial vehicle (UAV), UAV aerial photography images are used to realize real-time detection of on-site working conditions by real-time tracking of the working status of the head working and other related parts of the pumping unit. Considering the real-time performance of working condition detection, this paper proposes a framework that combines You only look once version 3 (YOLOv3) and a sort algorithm to complete multi-target tracking in the form of tracking by detection. The quality of the target detection in the framework is the key factor affecting the tracking effect. The experimental results show that a good detector makes the tracking speed achieve the real-time effect and provides help for the real-time detection of the working condition, which has a strong practical application.

Keywords: computer vision; oil well working condition; real-time detection; sort; unmanned aerial vehicle (UAV); YOLOv3

1. Introduction

The fault diagnosis technology and working condition monitoring technology of the pumping unit have always been the focus of the oilfield. At present, the commonly used fault diagnosis methods are mainly manual analysis and indicator diagram diagnosis. However, the dependence of a large number of high-precision sensors and high-sensitive devices not only increases the original cost of working condition detection but also gradually increases the requirements of staff [1]. The whole process takes a lot of time, and even real-time working conditions cannot be obtained. This has posed a great challenge to the detection of field working conditions of oil field pumping wells [2]. In recent years, with the gradual maturity of UAV technology, more and more projects have been launched around UAV, and it has been widely used in the inspection of power, highway, agriculture, communication, oil, and other fields [3]. By making use of the flexible mobility and powerful timeliness of the UAV, the difficulty of traditional condition detection can be overcome by using the UAV patrol mode [4,5].

The subject of this paper is the fine inspection research of pumping-well working conditions based on UAV. Unmanned UAVs equipped with high-definition cameras can hover in the air for a long time

to monitor the ground over a wide range and obtain real-time images. Therefore, through the pumping unit's real-time images acquired by the UAV, the deep learning detection [6,7] and the tracking method are used to detect the working condition of the oil-well pumping unit in operation. The specific detection precision is to the extent of the pumping unit's key parts [8]. At the same time of the whole pumping unit detection, the head working part of the pumping unit also undergo detailed detection and tracking, so as to achieve more refined inspection and get a more detailed pumping condition. Tracking the working state of the pumping unit and key components provides real-time position and movement information of the specified target [9]. By analyzing the state of the pumping unit and the real-time working state of the key components, the purpose of the drone's refined detection of the oil-well pumping unit is achieved [10].

Because there are multiple targets on the oil field, such as vehicles and workers, the purpose of this paper is to track multiple specified targets in the UAV image, which becomes a problem of multi-target tracking [11]. Multi-target tracking lacks artificial markers, and there are multiple targets, so it is necessary to use a target detector to detect the position of the target in the image at each moment [12]. Therefore, this paper adopts the tracking method based on detection and matching. Firstly, the detector is used to detect the static image of the oil-well pumping unit and the important parts, such as head working. Then, the static problem is extended to the dynamic problem, and the detection results of the two frames before and after are matched one by one to realize the tracking of the key working parts of the oil-well pumping unit and the pumping unit.

In this article, the main contributions are as follows. 1) A multi-target tracking framework (YLTS) for real-time tracking is proposed. It uses YOLOv3 as the detector and the sort algorithm as the tracker. In this paper, different algorithms are used as detectors to make multi-target tracking experiments for oilfield pumping units and their related components, and their accuracy and real-time tracking effects are compared. It is concluded that the use of YOLOv3 as the detector in this framework is most suitable; 2) different from the traditional method of detecting pumping unit working conditions with indicator diagrams, this paper applies the fine inspection project of UAV to the study of pumping unit working condition detection in oil production. By detecting and tracking the pumping unit and the head working part in the oil field, the position and movement information of the key components such as the head working are obtained, which provides a reliable basis for the next semantic analysis and the judgment of the working condition. so as to obtain the real-time working condition of the oil-well pumping unit.

The rest of this article is arranged as follows. Section 2 briefly reviews the research status of pumping unit working condition detection and the application status of UAV inspection. Then the related work of the model is introduced. The proposed method is described in Section 3, and experimental results and comparisons are explained in detail in Section 4. Finally, we summarize the paper and illustrate the future work in Section 5.

2. Related Works

The pumping unit has many major components, and the common faults are also complicated. In order to meet different fault inspections, the current pumping inspection methods generally involve manual collection. High-precision sensors and high-sensitivity devices are used to detect the load and displacement, current, voltage, stroke, and stroke parameters of the pumping unit. Then display the parameter values and the indicator diagram on the LCD screen. Although this method basically satisfies the basic needs of oilfields for pumping-unit monitoring, as the scale of oilfield mining is getting larger and larger, the establishment of this system is more and more difficult and expensive.

In recent years, UAVs have been widely used in the field of inspection. However, so far, the more mature inspection application of UAVs only stays in the inspection of pipelines and routes, such as highways, high-voltage power lines, and oil and natural gas pipelines. The UAV flies along the pipeline to be inspected. In the automatic flight mode, the built-in high-definition camera is used to point at the pipeline to be inspected to collect the image of pipeline details, which is then transmitted to the

ground station through wireless remote real-time transmission. In this paper, the application of UAV inspection is extended to the fine inspection of the working condition of the oil field pumping unit, so as to obtain the position of the pumping unit and the motion information of key parts in the video sequence in the middle and low altitude flight, providing a basis for further semantic layer analysis (motion state recognition, scene recognition, etc.) [13]. In this way, the real-time working condition of the oil-well pumping unit can be further judged according to the obtained information.

In order to achieve the work status tracking for pumping units key component, based on the requirement of real-time and multi-target, the technology adopted in this paper is the target tracking algorithm based on detection and matching. The detection quality in this method largely affects the tracking effect, so the key technology of this algorithm lies in the image target detection algorithm of deep learning. This chapter mainly introduces the main algorithms and related concepts used in this paper, including the principle of convolutional neural networks (CNN) in deep learning and the most advanced algorithms in the field of image detection, and time series prediction algorithms.

2.1. The Basics of Convolutional Neural Networks (CNNs)

The convolutional neural network (CNN) is a deep learning algorithm, which is an application of deep learning algorithms in the field of image processing and has excellent performance for large-scale image processing [14]. Inspired by the biological neural network, the perception layer was used to simulate the process of obtaining image information in biological vision, the hidden layer was used to simulate the neurons in the biological neural network, and the convolutional layer and excitation function were used to simulate the process of information transmission between neurons in the biological neural network. CNN uses a large number of hidden nodes to store the data of the original image. This method can obtain a better representation than the original image, and the tile processing method of hidden layer nodes makes the CNN have translation invariance. The schematic diagram of a CNN is shown in Figure 1:

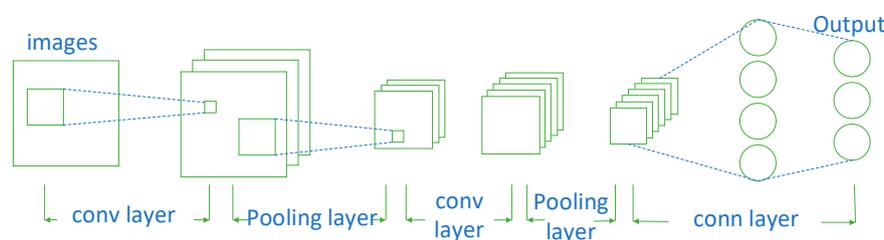


Figure 1. The basic construction of a convolutional neural network (CNN).

As shown in Figure 1, a CNN is made up of several convolution layers [15], a pooling layer, and a fully connected layer. Multiple convolutional layers are accompanied by a pooling layer. After repeated cycles, a fully connected layer is added to form a CNN. The convolution layer is the layer responsible for the transformation from the real domain to the feature domain, and it is also the most critical layer. The purpose of the pooling layer is to subsample the convolution result [16], extract the important part of the feature, reduce the number of network parameters, prevent the emergence of an over-fitting image, and improve the robustness of the network. The fully connected layer is mainly used to make some local features have global characteristics. All neuron nodes in this layer will be connected with the output of all neurons in the convolution layer of the previous Layer. Therefore, the calculation amount of the fully connected layer is relatively large. The output result of the fully connected layer will be taken as the input of the classifier [17].

2.2. Object Detection

Object detection refers to detecting the location of objects in an image while classifying images. The deep convolutional neural network (DCNN) has made great achievements in image object

detection after face recognition. In recent years, a large number of efficient object detection algorithms based on deep learning have emerged successively, such as the region-convolutional neural network (R-CNN), fast region-convolutional neural network (Fast R-CNN), faster region-convolutional neural network (Faster R-CNN), You only look once (YOLO), and Single Shot Multi-Box Detector (SSD) [18]. These algorithms are divided into two categories according to whether there is a region proposal.

2.2.1. Faster R-CNN

Faster R-CNN is the most advanced algorithm for object detection in R-CNN series images based on deep learning. It introduced the region proposal network (RPN) to directly generate candidate regions, which can be seen as a combination of the RPN and Fast R-CNN model [19].

For the RPN, a CNN model (commonly known as a feature extractor) is used to receive the whole picture and extract the feature graph. An $N \times N$ sliding window is then used on the feature graph to map a low-dimensional feature (e.g., 256-d) for each sliding window position. This feature is then fed into two fully connected layers, one for classification prediction and one for regression. For each window position is a set k different size or scale of a priori box (anchors, default bounding boxes), which means that each location has a prediction k candidate region (region proposals). For the classification layer, its output size is $2k$, which represents the probability value that each candidate region contains object or background, while the regression layer outputs $4k$ coordinate values, which represents the position of each candidate region (relative to each prior box). The two full connection layers are shared for each sliding window location. Therefore, RPN can be realized by convolution layer: firstly, an $n \times n$ convolution to obtain low-dimensional features, and then two 1×1 convolutions for classification and regression, respectively. The network architecture of RPN is shown in Figure 2.

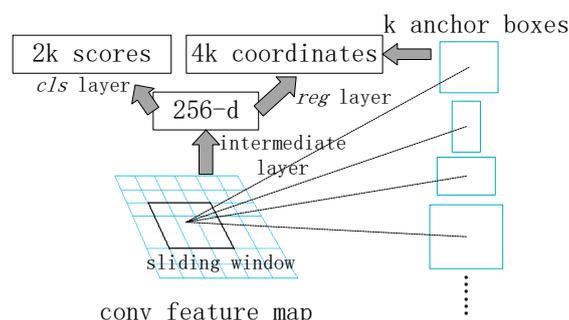


Figure 2. Region proposal network (RPN) network architecture.

The region proposal network uses dichotomies to distinguish only the background and objects but does not predict the categories of objects, namely class-agnostic. This method solves the regional recommendation and time-consuming problems in Fast R-CNN and greatly improves the detection speed. The detection of the mean average precision (mAP) value of PASCAL VOC 2007 increased from 70% to 73.2%.

2.2.2. YOLO

The YOLO neural network is based on the regression method to complete the target detection instead of the regional recommendation. It was proposed by Ross et al. in 2015 [20], which mainly transforms the multi-classification problem into a regression problem to solve the image detection. The classification and localization problems are solved by the same regression algorithm, which greatly improves the detection speed and achieves real-time effects in the field of general image target detection. YOLO first divides the whole picture into $S \times S$ grids. Each grid is responsible for predicting the position of the target point where the center point falls in this grid area. The predicted value is compared with the real value to calculate the predicted loss. The core idea is to directly operate on the entire picture, input a picture, and directly derive the position of the prediction frame and the category

to which the prediction frame belongs in the output layer. Each grid into which YOLO is divided is responsible for predicting some detection frames. Each detection frame needs to have a confidence value of a specific target in addition to its own position information.

By means of direct regression of the whole graph, YOLO can greatly improve the detection speed, reduce the error rate of background prediction, and learn highly generalized features, which is better than Fast RCNN in migration learning. However, the disadvantage is that the detection accuracy is low, object positioning errors easily occur, and the detection effect on small objects is not good enough. A series of YOLO algorithms have appeared (e.g., YOLOv2, YOLOv3) in recent years and have improved and strengthened the shortcomings of the original version. Based on the research of this paper focusing on real-time and multi-objective features, the detection part used in this paper is the latest YOLOv3 neural network in this series. The use of YOLOv3 neural network algorithm modeling to implement the detector portion of this article will be described in detail in Section 3.

2.2.3. SSD

The Single Shot Multi-Box Detector (SSD) belongs to the multi-box prediction of a one-stage method. The main idea is to carry out dense sampling uniformly on the feature graph of multiple layers in the image [21]. Different scales and aspect ratios can be adopted in sampling, and then features can be extracted by CNN for classification and regression. The whole process only takes one step, so it has the advantage of fast speed. However, an important disadvantage of uniform dense sampling is that training is difficult, mainly because the positive sample and the negative sample (background) are extremely unbalanced, resulting in slightly lower accuracy of the model [22].

Given the advantages and disadvantages of the RCNN series and the YOLO series, the SSD algorithm borrows many of these ideas and has many ideological improvements. Respectively, they are:

1. Multi-scale feature graph is adopted for detection—pyramid feature.
2. Set Default boxes.
3. Determination of Default boxes size.
4. Convolution was used for detection.

The above improvements made the detection speed faster than YOLOv1 and the accuracy faster than Faster R-CNN. However, the initial size and aspect ratio of the default boxes need to be set manually, and the size and shape of the default box used by the feature of each layer in the network are just different, which makes the debugging process very dependent on experience [23]. Moreover, the recognition of small-size objects is still poor, which cannot reach the level of Faster R-CNN. In contrast, the YOLOv3 used in this paper has obvious advantages in small object detection after absorbing the advantages and disadvantages of the first two versions and is much faster than SSD. This is one of the reasons why this article uses YOLOv3 instead of SSD as a detector.

3. Using the YLTS Framework to Realize the Pumping Unit Working Condition Detection of the Aerial Image of the UAV

This paper uses the proposed YLTS framework to achieve multi-target tracking [24,25]. Before the tracking, YOLOv3 was used to complete the detection of all the pumping units and the head working parts in the video to realize feature modeling, and then, the sorting tracking algorithm to complete the multi-target tracking was used. The whole process was to achieve multi-target tracking by detecting and then using prediction and matching. The framework proposed in this paper achieves real-time tracking, but mainly depends on the performance of the detector in the framework. YOLOv3, as a target detector, was a relatively good model in recent years. After experimental comparison, it is concluded that the use of YOLOv3 as a detector enables the framework to achieve faster real-time effects in tracking speed. Because the state of the UAV is in cruise, the main purpose of this article concerns the low altitude cruise in the detection of the oil pumping unit and the head working, and mainly discusses the work condition of the head working (work cycle, movement speed, movement direction)

for real-time tracking, access to the above information can be used according to its working status for further analysis of the pumping unit working condition.

3.1. Using YOLOv3 as a Detector of the YLTS Framework to Detect the Pumping Unit and the Head Working

In order to learn more about YOLOv3, the first two versions of YOLO (v1, v2) must be understood first. Since many of YOLOv3's ideas are inherited from v1 and v2, this section first introduces YOLOv1, and then introduces YOLOv3 in detail.

The earliest version of the YOLO series is YOLOv1, which is a detection model that converts multiple classification problems into regression problems for solution. The classification and location problems in the detection of a pumping unit and head working are solved by the same regression algorithm, which greatly improves the detection speed. It uses a separate CNN model to realize end-to-end target detection, divides the input images into 7×7 grids, and then each cell is responsible for predicting the targets in which the center points fall in the grid; when the pumping unit or head working fall in some grid, this grid is responsible for predicting them, compares the predicted value with the real value, and calculates the predicted loss. The core idea is to directly manipulate the whole picture by inputting a figure directly in the output layer for each grid to predict the B bounding box location information and the confidence score of the bounding box [26].

The predicted value of each bounding box contains five elements: (x,y,w,h,c) , where (x,y) represents the center coordinate of the boundary box, and the predicted value (x,y) of the center coordinate is the offset value relative to the coordinate point in the upper left corner of each cell; w and h are the width and height of the bounding box, and the predicted values of w and h of the bounding box are the ratio of the width and height relative to the entire image, and the value c is confidence score. The confidence score includes two aspects: on the one hand, the probability of the boundary box containing the target is denoted as $Pr(object)$; if the pumping unit or the head working part in the picture falls in the grid cell, it is set as 1, otherwise, it is 0. On the other hand, the accuracy of the boundary box can be represented by the intersection ratio (IOU) of the prediction box and ground truth, denoted as IOU_{pred}^{truth} , so the confidence is defined as $Pr(object) * IOU_{pred}^{truth}$. The multiplication of confidence scores and conditional probability is the solution of the classification problem, such as Formula (1):

$$Pr(class_i|object) * Pr(object) * IOU_{pred}^{truth} = Pr(class_i) * IOU_{pred}^{truth}. \quad (1)$$

As shown in Formula (1), it represents the confidence of the category. In the classification problem, each grid unit also predicts C conditional category probabilities $Pr(Class|Object)$ that are conditional on the inclusion of the target grid unit. Each grid cell predicts only one set of category probabilities, regardless of the number of bounding boxes B. This paper aims to detect the pumping unit and head working parts in the field with complicated environmental conditions, so C here is set as 2, which also reduces the workload of the algorithm. In the test, the conditional class probability is multiplied by the predicted confidence value of each box, so as to calculate the class-specific confidence scores of each boundary box, what it represents is the probability that the target belongs to a pumping unit or head working in the boundary box and the quality that the boundary box matches the target. Prediction boxes of the network are generally filtered according to category confidence. In general, each cell needs to predict $(B \times 5 + C)$ values. If the input image is divided into an $S \times S$ grid, the final predicted value is a tensor of $S \times S \times (B \times 5 + C)$ size.

The structure of the YOLO network can be seen from Figure 3 [27], which uses the convolutional network to extract features and then uses the full connection layer to obtain predicted values. It can be seen that its network has 24 convolutional layers and two fully connected layers. The fully connected layer of the last layer outputs a $7 \times 7 \times 30$ tensor; this tensor stores the location information of all the detection boxes predicted by the YOLO model and the probability values that belong to a set of specific classes with the detection boxes.

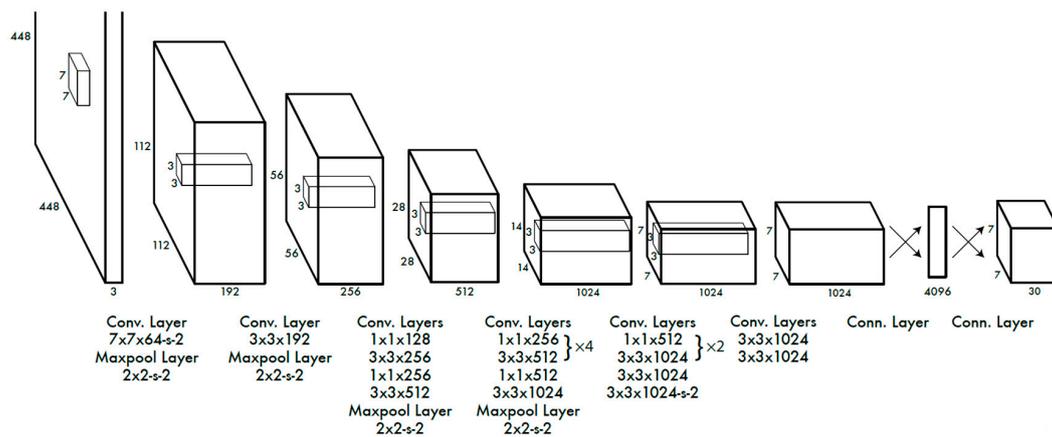


Figure 3. The YOLOv1 network structure.

The training of YOLO is end-to-end, the prediction of the position, size, type, confidence (score), and other information of the prediction box is trained by a loss function [28]. Formula (2) is YOLOv1’s loss function.

$$\begin{aligned}
 loss = & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B l_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\
 & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B l_{ij}^{obj} \left[(\sqrt{\omega_i} - \sqrt{\hat{\omega}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \\
 & \sum_{i=0}^{S^2} \sum_{j=0}^B l_{ij}^{obj} (c_i - \hat{c}_i)^2 + \\
 & \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B l_{ij}^{noobj} (c_i - \hat{c}_i)^2 + \\
 & \sum_{i=0}^{S^2} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2.
 \end{aligned} \tag{2}$$

The S^2 in Formula (2) represents the number of grids, in this case, 7×7 . B is the number of prediction boxes per cell, in this case, is 2. The value of l_{ij}^{obj} is 0 or 1, that is, whether there is a target in the cell. The value of λ_{coord} is 5 and the value of λ_{noobj} is 0.5. Formula (2) is divided into four parts:

- Part 1: The first line is the loss function for position prediction. The total square error (SSE) is used.
 - Part 2: The second line is the loss function for width and height. The total square error is used.
 - Part 3: The third and fourth rows of confidence (confidence) are also the total squared error (SSE) used as a loss function.
 - Part 4: The fifth line is the loss function for the class probability and also uses the total square error (SSE) as the loss function.
- Finally, several loss functions are added together as a loss function of YOLOv1.

Different oilfields have different environmental conditions. In the complex environment of oilfields, the pumping unit is connected to the head working. In addition, the head working is relatively small compared with the pumping unit when the UAV is flying higher. Moreover, the up and down swing of the head working in the pumping unit may lead to the overlap with the pumping unit itself. Under such complex and harsh testing conditions, YOLOv1 cannot meet the requirements of the industrial application level. YOLOv3’s improvements make it an algorithm that meets the requirements of the industrial application level requirements. On the surface, the core idea of YOLOv3 is basically the same as that of YOLOv1, both of which are tested by dividing cells in a square way, but the number of partitions is different. However, its improvement makes its detection effect become an excellent detector both for accuracy and speed. For example, batch normalization has been added since v2 as a method of regularization, accelerating convergence, and avoiding overfitting, connecting the BN layer and leaky

ReLU layer to the end of each convolutional layer. The use of multilevel prediction makes up for the shortcomings of the previous version of small target detection. Multi-scale training, which allows for a trade-off between speed and accuracy, makes YOLOv3 more flexible and suitable for industrial applications. Figure 4 shows the network structure of YOLOv3.

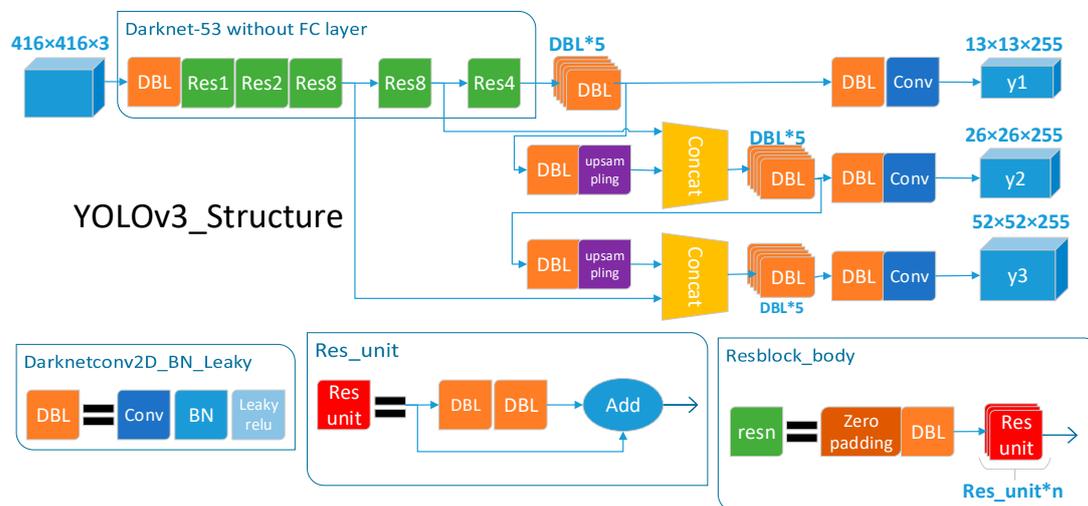


Figure 4. The YOLOv3 network structure.

Here are three additions to Figure 4:

First of all, DBL is the basic component of YOLOv3, which consists of convolution, BN, and Leaky Relu. For v3, in addition to the last layer of convolution, the three have been merged to form the smallest component. Secondly, there are multiple res, which are the big components of YOLOv3. They draw on ResNet's residual structure. Using this structure can make the network structure deeper. Its basic component is also DBL. Finally, splicing the intermediate layer of darknet and the upper sampling of a later layer. The splicing operation is different from the residual layer add operation. Splicing expands the dimension of a tensor, whereas add simply adds without changing the dimension of a tensor.

There is no pooling layer and full connection layer in the entire v3 structure, add an anchor box to predict the bounding box. This avoids the image that can only recognize the same resolution as the training image at the time of detection and can have a higher resolution at the output of the convolutional layer. It is very suitable for the occasion when the UAV is not in the fixed altitude inspection. Good detection can be maintained when the drone's flight is very close to a pumping unit or the flight altitude is high. In the process of forward propagation, the dimensional transformation of the tensor is realized by changing the step size of the convolution kernel [29]. The following analysis is carried out layer by layer.

Input layer: images are input with 416×416 pixels and 3 channels, and then the BN operation is carried out on the input. Then, the 32-layer convolution kernel operation is carried out. The size of each convolution kernel is 3×3 , and the step is 1. Finally, the 416×416 feature map of 32 channels is produced as the output.

Res layer: the input and output in this layer are generally consistent, and no other operations, just subtraction. In order to solve the phenomenon of gradient diffusion or gradient explosion in deep neural network, it is proposed to change layer by layer training to stage by stage training. The deep neural network is divided into several subsegments, each of which contains a relatively shallow network layer, and then each segment is directly connected to train the residual. Each segment learns only a fraction of the total difference, and ends up with a smaller total loss. At the same time, the propagation of the gradient is well controlled to avoid situations that are not conducive to training, such as the disappearance or explosion of the gradient.

Darknet-53: from layer 0 to layer 74, there are 53 convolution layers, and the rest are res layers. This layer is the main network structure for feature extraction of YOLOv3, and the convolution layer of 3×3 and 1×1 is used [30]. A large number of jump layer connections using residuals. In the previous work, the sampling was generally conducted by max-pooling or average-pooling with the size of 2×2 and stride length of 2. However, in this network structure, convolution with a step size of 2 is used for descending sampling. At the same time, up-sampling and route operation are used in the network structure, and three times of detection are carried out in a network structure. This ensures the convergence of training. The effect of classification and detection will also be improved, and the reduction of parameters will reduce the amount of calculation. This is very good for more complex oil field sites, different locations of the sparse distribution of pumping units, plus the blocking of the head working. Better results can be obtained by using a darknet-53 network to train such complex images.

The part of YOLO: this part is divided into three scales from 75 to 105 layers, and local feature interaction is realized by a convolution kernel, such as in Figure 5.

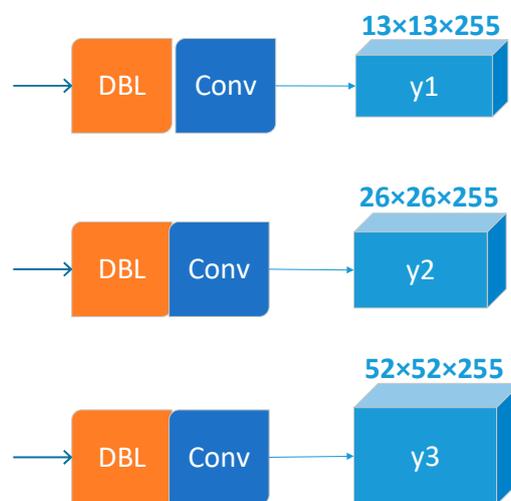


Figure 5. Output from the YOLO layer.

The minimum scale YOLO layer inputs 13×13 feature maps, a total of 1024 channels, reduces the channel to 75 by convolution operation, and finally outputs 13×13 feature maps and 75 channels, and on this basis, perform position regression and classification.

The input of the mesoscale YOLO layer is to convolve the feature map of the 13×13 and 512 channels of the 79 layer to generate the feature map of 13×13 and 256 channels. A 26×26 , 256-channel feature map is generated after up sampling, and convolution is performed after merging with the 61×26 , 512-channel mesoscale feature map of the 61 layer. Finally, an output of a 26×26 size feature map and 75 channels is produced.

The input of the large-scale YOLO layer is to convolve the feature map of the 91-story 26×26 and 256-channel and generate the feature map of 26×26 and 128 channels and generate the feature map of 52×52 and 128 channels after up sampling. At the same time, convolution is performed after merging with the 52×52 , 256-channel mesoscale feature map of 36 layers. Finally, a feature map of size 52×52 and 75 channels are output. Based on this, position regression and classification are performed [31].

According to the structural pattern of YOLOv3, except for the last layer of the model, which uses the linear activation function, all other layers use the leaky ReLU below as the activation function:

$$y = \begin{cases} x, & x > 0 \\ 0.1x, & otherwise \end{cases} \quad (3)$$

Compared to ordinary ReLU, leaky does not make the negative number directly 0, but multiplies it by a small coefficient (constant). Keep negative output, but reduce negative output.

Compared with YOLOv1, v3 makes some adjustments in the loss function. Except that the loss function of the width and height of the second part still uses the total square error, the loss function of other parts uses the binary cross entropy. The next step is to add them together. The loss function for V1 was explained in Formula (2) in the previous section. The following is the formula for binary cross entropy:

$$loss = - \sum_{i=1}^n \hat{y}_i \log y_i + (1 - \hat{y}_i), \quad (4)$$

$$\frac{\partial loss}{\partial y} = - \sum_{i=1}^n \frac{\hat{y}_i}{y_i} - \frac{1 - \hat{y}_i}{1 - y_i} \quad (5)$$

This is the loss function between probabilities. Only when y_i and \hat{y}_i are equal, the loss will be 0; otherwise, the loss will be a positive number. Moreover, the greater the difference in probability, the greater the loss will be. This measure of probability distance is called cross entropy. YOLOv3 changes the loss function so that it can better model complex target categories and data sets of overlapping labels. It is also suitable for the data set that the head working overlaps or blocks with the pumping unit in the scene of the oil field in this paper.

Through the above modeling, the work of the detector is first completed. The detection of each frame of the pumping unit and the head working part is realized. After that, the tracker is used to complete the tracking of multiple targets.

3.2. Use the Sort Algorithm as a Tracker of the YLTS Framework to Track the Pumping Unit and the Head Working

In order to ensure the real-time tracking effect, this paper uses the Sort algorithm as a tracker to track the target based on the detector's detection of the pumping unit and the head working. The algorithm is an algorithm based on detection and multi-target tracking, which is updated online and has good real-time performance. The tracking problem is regarded as a data association problem. The Kalman filter is used to process the correlation of frame-by-frame data [32,33], and the Hungarian algorithm is used to correlate metrics. The position and size of the detection box are used to correlate the motion estimation and data of the target [34]. The following is an object state model that represents and propagates the target ID to the next frame:

$$x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T. \quad (6)$$

where u and v represent the central coordinate of the target, s represents the size area of the target, r represents the aspect ratio of the target, which remains unchanged, and the last three quantities represent the predicted next frame.

The steps of the whole process are as follows:

1. When the first frame comes in, the detected target is initialized and a new tracker is created, labeled with an id.
2. When the following frame comes in, the state prediction and covariance prediction generated by the previous frame detection box are obtained first in the Kalman filter. The target state prediction and the IOU of the frame detection box are respectively obtained, the maximum matching of the IOU is obtained by the Hungarian assignment algorithm, and the matching pair in which the matching value is smaller than the IOU threshold is removed.
3. The Kalman tracker is updated using the matched target detection frame in this frame to calculate the Kalman gain, status update, and covariance update. The status update value is output as the tracking frame of this frame. The tracker for targets that are not matched in this frame are reinitialized [35,36].

After the above steps, the proposed YOLOv3 is used as the detector, and the sort algorithm is basically completed as the framework of the tracker. First, use YOLOv3 to test the pumping unit and the head working part, and input the test result to the tracker. As a tracker, the sort algorithm uses the Kalman filter to process the correlation of frame-by-frame data and the Hungarian algorithm to correlate metrics to track the pumping unit and the head working. After that, through the analysis of the results of the tracking, the real-time working condition of the pumping unit can be obtained.

4. Experiment and Analysis

In this paper, the UAV is used for video capture, and the video is processed by frame separation. The image marking tool is used to mark the pumping unit and the head working, and the training data set is produced. The Tensorflow-GPU [37] version is used as a framework for deep learning, implemented under the Linux operating system, using 1080Ti GPU for image training and target detection and tracking in the video. The detection speed and mAP value are used to analyze the advantages of the YOLOv3 algorithm as a detector in the framework proposed in this paper, to achieve a good real-time tracking effect, and make decisions for the detection of the working condition.

4.1. Description of the Training Data

In this paper, UAV aerial photography inspection data provided by China Petroleum Western Drilling Engineering Co., Ltd., were screened through screening and editing to select 5 videos for 130 min with a resolution of 640×480 . Four of them are medium and low altitude flight (15–25 m), and one video is high altitude flight (45–55 m). After the data from three videos were processed by interval frames, the parts of the data that did not meet the training conditions were removed. The training set contained about 5400 images, and the data from the remaining two videos were processed by interval frames as the test set images, with about 2500 images. A part of the data set is shown in Figure 6.



Figure 6. Part of the image in the dataset.

After the video data was processed in a frame-by-frame process, the pumping unit and the head working were manually labeled using an image labeling tool. After each image was annotated, a class label file was generated, which stores the position of the label box and the category information, as shown in Table 1:

Table 1. The examples of annotated data.

Class ID	Normalization of the Central Point x Value	Normalization of the Central Point y Value	Normalization of w Value	Normalization of h Value
0	0.6698369565217391	0.4565916398713826	0.34148550724637683	0.9131832797427653
0	0.3675781250000003	0.6263888888888889	0.20390625	0.4666666666666667
0	0.3583333333333334	0.5857142857142857	0.46	0.5428571428571429
1	0.5083333333333334	0.4768356643356643	0.6266666666666667	0.7159090909090909
1	0.6391666666666667	0.65	0.22166666666666668	0.3666666666666667
1	0.6216666666666667	0.43214285714285716	0.41000000000000003	0.7214285714285715

It can be shown from Table 1 that the class label with ID 0 is the pumping unit, and the class label with ID 1 belongs to the head working. The center point x,y coordinate value, the width value w, and the height value h of the detection frame are all normalized according to the image size.

The purpose of frame separation processing is to improve the processing speed of the whole system without affecting the prediction ability of the Kalman filter. The direct impact of video frame separation processing is whether the target position change rate can be learned. If the interval is too long, there will be a phenomenon in which the image information is not extracted when the target trajectory changes greatly, which will lead to an unstable change of the learned position. In order to verify the effect of different interval frames on the learning ability of the frame on the target trajectory, each video in this paper trains the network at intervals of 1, 5, 10, 15, and 25 frames and tests them. Figure 7 shows the average of the loss results of the three videos of the training set after different interval frame numbers.

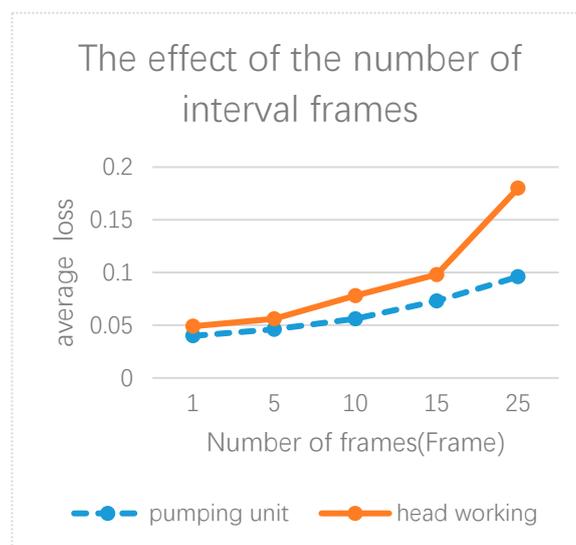


Figure 7. Comparison of average loss results at different intervals.

As can be seen from Figure 7, there is a small gap between different interval frames in the prediction of the target trajectory in the video total, especially in the case of small intervals, but when the interval frames are too large, the prediction ability will decline sharply. It can be concluded from the results that the Kalman filter can learn the target motion rule well. However, when the frame interval time is larger, the target motion regularity is weaker, and the prediction effect will be worse. Because the pumping unit is in a working state during the inspection of the drone, the head working is often obscured or incomplete. Therefore, the data of head working in the obtained data is relatively poor compared to the overall pumping unit. When the number of interval frames is large, the loss result will also be worse than that of the pumping unit.

4.2. Experimental Results and Comparison

The tracking framework used in this article makes the tracking effect dependent on the quality of the detector; therefore, different detectors are used in this paper to make comparative experiments. SSD is an algorithm similar to YOLOv3 in performance and core thinking; thus, the comparison of detectors in the following section is mainly to compare SSD with YOLOv3. Figure 8 shows the detection and tracking effect of a single target. The blue box is the detection box, and the white one is the tracking box. The purpose of the detection box is to accurately find the location and size of the target to be found in each frame and mark it out. The tracking box relies on the detection box to match the detection box before and after the frame and to predict the motion and similarity of the tracking target. For the occluded target, the detection box will not appear, because there is no target to be detected in the image. In this paper, for the occluded target in a short time, the tracking box will continue to track it according to the prediction in the previous frame.



Figure 8. Single target effects.

Whether it is SSD or YOLOv3 as a detector, the detection of a single target can get better results. Although the pumping unit has no movement change, with the movement of the UAV, the detection box and tracking box can accurately follow the target. Moreover, the box of the head working can also move as it moves up and down.

In the case of a medium or low flight height of the UAV, different algorithms are used as detectors to detect and track multiple targets, which are shown in Figures 9 and 10.



Figure 9. The tracking effect with Single Shot Multi-Box Detector (SSD) as the detector.



Figure 10. The tracking effect with YOLOv3 as the detector.

It can be shown from Figure 9 that a pumping unit in the lower left corner was not detected, which also led to the failure of tracking, while the tracking of YOLOv3 as a detector succeeded. However, neither achieved a tracking effect on targets with long-term obscuration, which is the sacrifice of the sort algorithm in this framework to achieve a faster tracking speed. However, the flexibility based on drones can make up for this shortcoming. In terms of speed, YOLOv3 as the detector is faster, which is also the advantage of the algorithm for the detecting speed. For the shadowing problem, this paper makes the following test to test the critical value of tracking failure.

As shown in Figure 11, with a test for the critical value of the tracking effect in the case of shielding, it can be seen that the four images are continuously intercepted while the head working of the left pumping unit is slowly leaving the video viewing angle. The head working in the first three pictures is still in the line of sight of the drone, but are slowly decreasing. Still, it can still be tested and tracked, and the last one shows that when the head working disappears completely in the line of sight, it immediately loses its detection and tracking effect. Moreover, there is also a pumping unit behind the pumping unit on the left side of Figures 1–3, but they are not detected and tracked because of the occlusion. This is because the sort tracking algorithm only uses the position and size of the detection frame to perform the motion estimation and data association of the target in pursuit of the tracking speed. When the target is lost, it cannot be found, and the ID can only be re-updated through detection. Therefore, the critical value in the case of occlusion is that the tracking effect is lost when the occlusion is completely occluded or the detector does not detect the target due to occlusion. However, this is when the target disappears in the entire image. In Figures 9 and 10, multiple pumping units are working side by side, which causes the pumping unit and head working to be obscured by other pumping units. When the obstacle is detected, the Kalman filter can predict the position of the object in the detection box at the next moment. However, this prediction is very rough. When the object appears again, it is tracked through matching. However, the frame proposed in this paper is exactly in line with the scene of the oil field, and the shielding time is almost zero. Moreover, the UAV is in the way of patrol inspection, which also increases the probability of avoiding shielding and reduces the shielding time.

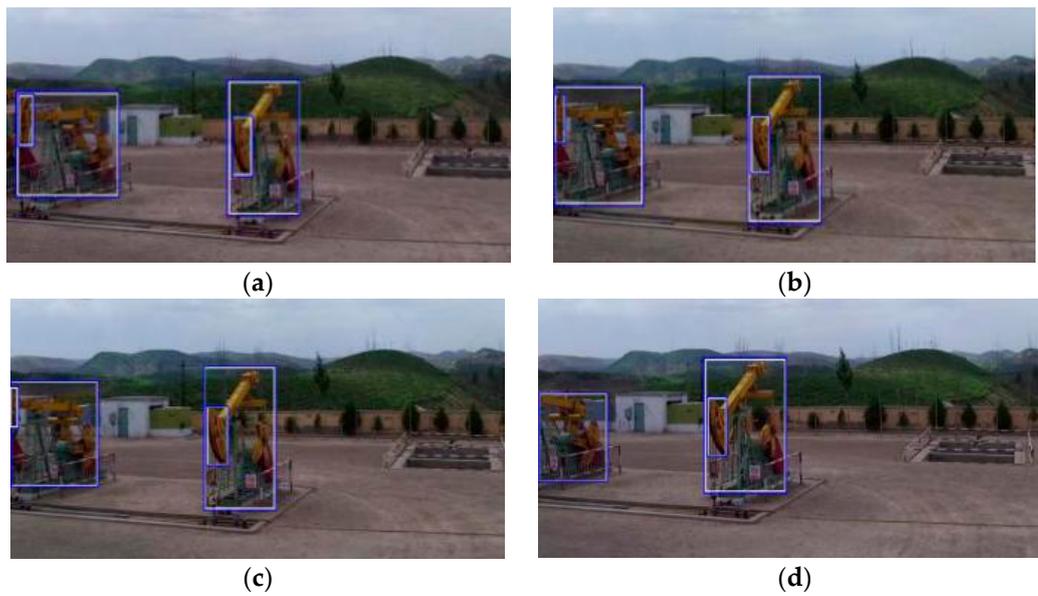


Figure 11. Masking of the critical value test. (a) Frame 180, (b) frame 185, (c) frame 188, (d) frame 196.

In the case of the UAV flying at a high altitude, different algorithms are used as detectors to detect and track multiple targets, which are shown in Figures 12 and 13.



Figure 12. The tracking effect with SSD as detector.



Figure 13. The tracking effect with YOLOv3 as detector.

It can be seen from Figures 12 and 13 that the detection and tracking effect of the pumping unit was achieved, but the tracking result with SSD as the detector did not detect and track the head working position. This is related to the performance of the detector. For YOLOv3, the defects of small targets that could not be detected in the previous series have been improved, so compared to SSD as

a detector, YOLOv3 has a better effect on detecting small targets. The tracking effect in this paper also depends on the quality of the detector, so it can be seen that the tracking effect with SSD as the detector does not track small targets.

4.3. The Analysis of Experiment

When training in the detector section, the default number of iterations for YOLOv3 training is 500,200. After 500,200 iterations, the training will stop automatically. Training can also be stopped when the loss is no longer falling or the drop is very slow. The training log should be saved after the training and the following loss curve drawn using python. In order to make the contrast more vivid, the training loss curve of the SSD is drawn by taking the iteration times and the same iteration interval of YOLOv3.

As shown in Figure 14, the training stops at 16,000 iterations, and the loss value finally converges to 0.05. In this experiment, since the average loss of YOLOv3 is very slow and substantially converged after less than 0.05, the threshold for stopping the training is set to 0.05 at the time of this training. When the loss value reaches 0.05, the number of iterations is about 16,000. Therefore, the training iteration of the SSD is also taken from the log between 6000 and 16,000 to draw Figure 15. The above two loss graphs show that YOLOv3 basically converges to 0.05, and after 16,000 iterations, the SSD's loss curve still fluctuates between 0.25 and 0.1 and does not converge. It can be concluded that YOLOv3 has the advantage of training. The loss curve is not only faster than SSD convergence but also has a smaller convergence value. Therefore, YOLOv3 is more suitable as a detector in this paper than the SSD algorithm.

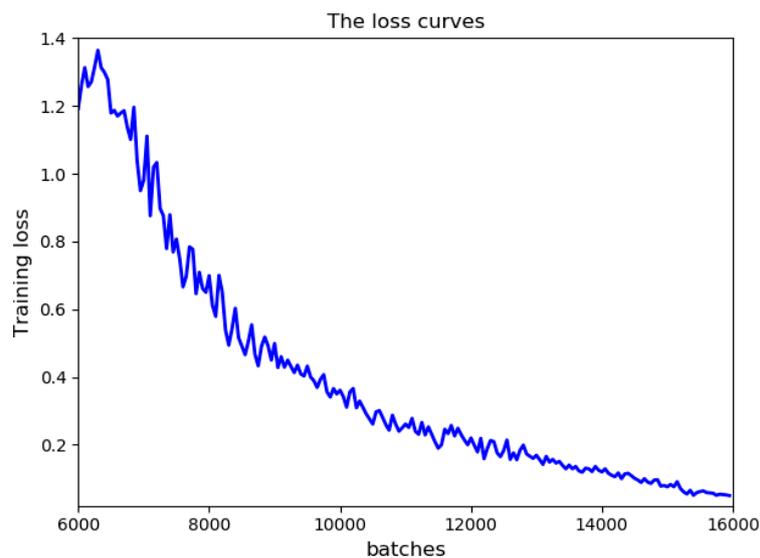


Figure 14. The loss curves of YOLOv3.

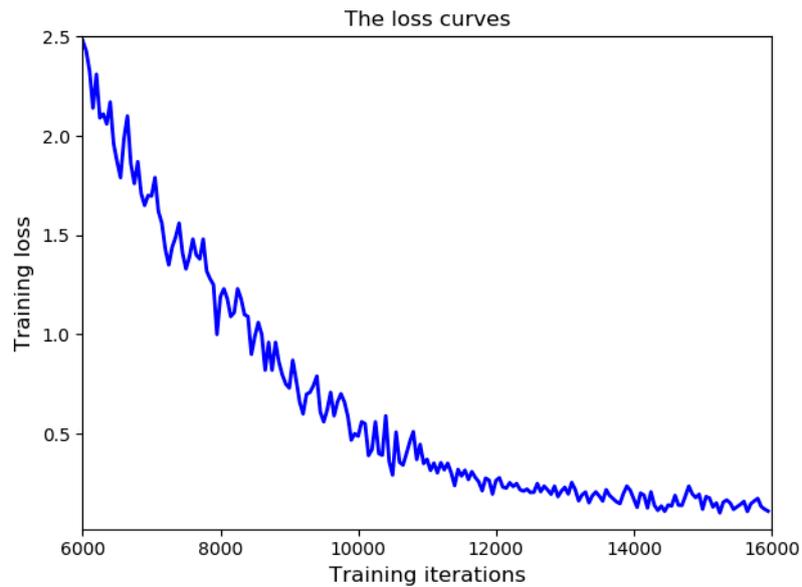


Figure 15. The loss curves of SSD.

In the experiment of this paper, the SSD algorithm with similar performance to YOLOv3 is compared with the algorithm used in this paper to compare the advantages of YOLOv3 as the detector in this paper. However, a comparison of the Faster R-CNN algorithm [38], the most advanced in object detection based on deep learning R-CNN series images, is also added. The comparison is mainly made from two aspects, the detection speed and mAP value. The former directly affects the real-time detection and tracking of the pumping unit and the head working, while the latter reflects the accuracy of detection and is the performance evaluation of the detector.

The mean average precision (mAP) is shown in Formula (7):

$$mAP = \int_0^1 P(R)d(R) \quad (7)$$

where P is the accuracy of the pumping unit and the head working, and R is their recall rate. The formulas for R and P are shown in Formulas (8) and (9), respectively:

$$P = \text{Number of targets detected} / \text{The total number of detected detection frames} \quad (8)$$

$$R = \text{The total number of detected targets} / \text{Verify the total number of all marked pumping units and the head working in the set} \quad (9)$$

As shown in Table 2, mAP values and the target detection speed of the three algorithms are respectively displayed.

Table 2. Test results for the three models.

Model	mAP(%)	Time for Detection(s)
Faster R-CNN	57.6	248
SSD	64.7	39
YOLOv3	64.5	20

It can be shown from Table 2 that YOLOv3 reached 64%; although the mAP of YOLOv3 is 0.02% less than that of SSD, it is almost the same. However, in terms of time, YOLOv3 only uses 20 s, which is much shorter than the time of the above two algorithms. It fully meets the requirements of real-time

performance emphasized in this paper. Therefore, it can be concluded that YOLOv3 is the most suitable detector for this experiment in terms of both accuracy and speed.

Finally, we compare the advantages and disadvantages of the proposed framework with other multi-target tracking algorithms, as this paper focuses on industrial applications, especially in this paper, for the tracking of oil field pumping units and head working. Therefore, the first two methods with the fastest processing speed of MOT Challenge2016 are selected for comparison. According to the size of the MOTA scores, the comparison results are shown in Table 3.

Table 3. The quality of evaluation of different methods.

Tracker	MOTA	MOPI	FP	FN	ID SW	HZ
YLTS	57.6	79.6	8698	63,245	1423	60.1
SMMUML	43.3	74.8	8463	93,892	985	187.2
LP2D	35.7	75.5	5084	111,163	1264	49.3

As shown in Table 3, the two algorithms with the fastest processing speed are compared with the framework proposed in this paper. The fastest algorithm is 182.7 HZ, which is far higher than all other algorithms. The processing speed of the framework proposed in this paper ranks second, which is more suitable for industrial applications, thanks to the processing speed of YOLOv3 and the sort algorithm. However, some other factors are sacrificed. IDSW is relatively high, which is also used to improve the speed and lead to more ID changes. Generally speaking, this framework achieves the second level in terms of processing speed on the premise of maintaining a high MOTA level. In combination with speed and accuracy, it can be seen that the proposed multi-target tracking framework has achieved good results.

5. Conclusions and Future Works

The Faster R-CNN, SSD, and YOLOv3 algorithms used in the experiments in this paper were used as detectors in the tracking framework proposed in this paper. The framework uses sort tracking to meet the real-time nature of the oilfield well conditions, which also puts the focus of this framework on the detector. The quality of the tracking depends entirely on the quality of the detector. Experiments have shown that YOLOv3 is the most suitable detector for this article, both in terms of accuracy and speed. However, the framework also has shortcomings. The detector and tracker used in this paper are designed to meet real-time performance, so it is faster in speed, but it also sacrifices tracking in special cases. For example, in the case of a long-term occlusion, the target being tracked will be lost, and the target ID will be frequently switched, which reduces the tracking effect. However, based on the background of the drone's refined inspection, this situation has also been reduced. Therefore, the final result can be used to track the pumping unit and the key components, such as head working, to obtain the position and motion information of the target, and to provide a basis for further semantic layer analysis (motion state recognition, scene recognition, etc.). In this way, the working conditions are checked in real-time.

According to the current research results, this paper believes that although the tracking target does not appear to be occluded for too long in the scene of drone inspection, it cannot ignore the existence of this situation. Considering the problem of target occlusion in the tracker is a concern for future research. This also reduces the dependence on the detector, reduces the number of ID switching during the tracking target, and improves the overall tracking performance. After obtaining the information of the tracking target, further motion analysis of the target working state to obtain clearer working conditions is also a concern for future research.

Author Contributions: Conceptualization, Y.Z., C.W., Q.W. and N.X.; data curation, Y.Z., Q.W. and Z.M.E.; formal analysis, Y.Z., C.W., N.X., Z.M.E. and S.Z.; funding acquisition, C.W.; investigation, Y.Z., Q.W. and S.Z.; methodology, Y.Z.; project administration, C.W., N.X. and S.Z.; resources, C.W., Q.W. and Z.M.E.; software, Y.Z.

and S.Z.; supervision, C.W., N.X. and S.Z.; validation, Y.Z.; visualization, Y.Z., Q.W. and Z.M.E.; writing—original draft, Y.Z.; writing—review & editing, Y.Z., C.W. and N.X.

Funding: This research was supported by Technology Innovation Action Plan Project (19511105103, 17511107203) and the National Key Research and Development Program of China (2018YFC0810204, 2018YFB17026) and National Natural Science Foundation of China (61872242), Shanghai Science and Shanghai key lab of modern optical system.

Acknowledgments: The authors would like to appreciate all anonymous reviewers for their insightful comments and constructive suggestions to polish this paper in high quality. Thanks to the data provided by China Petroleum West Drilling Engineering Co., Ltd. to support this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, H.; Wu, C.; Huang, W.; Wu, Y.; Xiong, N. Design and Application of System with Dual-Control of Water and Electricity Based on Wireless Sensor Network and Video Recognition Technology. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718795951. [[CrossRef](#)]
- Li, X.; Zhou, C.; Tian, Y.-C.; Xiong, N.; Qin, Y. Asset-Based Dynamic Impact Assessment of Cyberattacks for Risk Analysis in Industrial Control Systems. *IEEE Trans. Ind. Inform.* **2018**, *14*, 608–618. [[CrossRef](#)]
- Ju, C.; Son, H. Multiple Uav Systems for Agricultural Applications: Control, Implementation, and Evaluation. *Electronics* **2018**, *7*, 162. [[CrossRef](#)]
- Hu, G.; Yang, Z.; Han, J.; Huang, L.; Gong, J.; Xiong, N. Aircraft Detection in Remote Sensing Images Based on Saliency and Convolution Neural Network. *EURASIP J. Wirel. Comm. Netw.* **2018**, *2018*, 26. [[CrossRef](#)]
- Hua, X.; Wang, X.; Rui, T.; Wang, D.; Shao, F. Real-Time Object Detection in Remote Sensing Images Based on Visual Perception and Memory Reasoning. *Electronics* **2019**, *8*, 1151. [[CrossRef](#)]
- Aksu, D.; Aydin, M.A. Detecting Port Scan Attempts with Comparative Analysis of Deep Learning and Support Vector Machine Algorithms. In Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, 3–4 December 2018. [[CrossRef](#)]
- Wu, C.; Luo, C.; Xiong, N.; Zhang, W.; Kim, T. A Greedy Deep Learning Method for Medical Disease Analysis. *IEEE Access* **2018**, *6*, 20021–20030. [[CrossRef](#)]
- Huang, H.; Xu, Y.; Huang, Y.; Yang, Q.; Zhou, Z. Pedestrian Tracking by Learning Deep Features. *J. Vis. Commun. Image Represent.* **2018**, *57*, 172–175. [[CrossRef](#)]
- Wang, S.; Wu, C.; Gao, L.; Yao, Y. Research on Consistency Maintenance of the Real-Time Image Editing System Based on Bitmap. In Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hsinchu, Taiwan, 21–23 May 2014. [[CrossRef](#)]
- Zhao, J.; Xu, H.; Liu, H.; Wu, J.; Zheng, Y.; Wu, D. Detection and Tracking of Pedestrians and Vehicles Using Roadside Lidar Sensors. *Transp. Res. Part C Emerg. Technol.* **2019**, *100*, 68–87. [[CrossRef](#)]
- Jiang, X.; Fang, Z.; Xiong, N.N.; Gao, Y.; Huang, B.; Zhang, J.; Yu, L.; Harrington, P. Data Fusion-Based Multi-Object Tracking for Unconstrained Visual Sensor Networks. *IEEE Access* **2018**, *6*, 13716–13728. [[CrossRef](#)]
- Liu, C.; Zhou, A.; Wu, C.; Zhang, G. Image Segmentation Framework Based on Multiple Feature Spaces. *IET Image Process.* **2015**, *9*, 271–279. [[CrossRef](#)]
- Yang, J.C.; Jiao, Y.; Xiong, N.; Park, D.S. Fast Face Gender Recognition by Using Local Ternary Pattern and Extreme Learning Machine. *TIIS* **2013**, *7*, 1705–1720.
- Xue, W.; Wenxia, X.; Guodong, L. Image Edge Detection Algorithm Research Based on the Cnns Neighborhood Radius Equals 2. In Proceedings of the 2016 International Conference on Smart Grid and Electrical Automation (ICSGEA), Zhangjiajie, China, 11–12 August 2016. [[CrossRef](#)]
- Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
- Wan, L.D.; Zeiler, M.; Zhang, S.; Lecun, Y.; Fergus, R. Regularization of Neural Networks Using Dropconnect. *Int. Conf. Mach. Learn.* **2013**, *28*, 1058–1066.
- Wang, Z.; Lu, W.; He, Y.; Xiong, N.; Wei, J. *Re-Cnn: A Robust Convolutional Neural Networks for Image Recognition*; Springer: Berlin, Germany, 2018.

18. Napiorkowska, M.; Petit, D.; Marti, P. Three Applications of Deep Learning Algorithms for Object Detection in Satellite Imagery. In Proceedings of the IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018. [[CrossRef](#)]
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
20. Lan, W.; Dang, J.; Wang, Y.; Wang, S. Pedestrian Detection Based on Yolo Network Model. In Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China, 5–8 August 2018. [[CrossRef](#)]
21. Wang, X.; Hua, X.; Xiao, F.; Li, Y.; Hu, X.; Sun, P. Multi-Object Detection in Traffic Scenes Based on Improved Ssd. *Electronics* **2018**, *7*, 302. [[CrossRef](#)]
22. Biswas, D.; Su, H.; Wang, C.; Stevanovic, A.; Wang, W. An Automatic Traffic Density Estimation Using Single Shot Detection (Ssd) and Mobilenet-Ssd. *Phys. Chem. Earth Parts A/B/C* **2018**, *110*, 176–184. [[CrossRef](#)]
23. Kitayama, T.; Lu, H.; Li, Y.; Kim, H. Detection of Grasping Position from Video Images Based on Ssd. In Proceedings of the 2018 18th International Conference on Control, Automation and Systems (ICCAS), Daegwallyeong, Korea, 17–20 October 2018.
24. Zhang, H.; Gao, L.; Xu, M.; Wang, Y. An Improved Probability Hypothesis Density Filter for Multi-Target Tracking. *Optik* **2019**, *182*, 23–31. [[CrossRef](#)]
25. Yang, T.; Cappelle, C.; Ruichek, Y.; El Bagdouri, M. Online Multi-Object Tracking Combining Optical Flow and Compressive Tracking in Markov Decision Process. *J. Vis. Commun. Image Represent.* **2019**, *58*, 178–186. [[CrossRef](#)]
26. Shinde, S.; Kothari, A.; Gupta, V. Yolo Based Human Action Recognition and Localization. *Proced. Comput. Sci.* **2018**, *133*, 831–838. [[CrossRef](#)]
27. Krawczyk, Z.; Starzyński, J. Bones Detection in the Pelvic Area on the Basis of Yolo Neural Network. In Proceedings of the 19th International Conference Computational Problems of Electrical Engineering, Banska Stiavnica, Slovakia, 9–12 September 2018. [[CrossRef](#)]
28. Liu, X.; Yang, T.; Li, J. Real-Time Ground Vehicle Detection in Aerial Infrared Imagery Based on Convolutional Neural Network. *Electronics* **2018**, *7*, 78. [[CrossRef](#)]
29. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple Detection During Different Growth Stages in Orchards Using the Improved Yolo-V3 Model. *Comput. Electr. Agric.* **2019**, *157*, 417–426. [[CrossRef](#)]
30. Tumas, P.; Serackis, A. Automated Image Annotation Based on Yolov3. In Proceedings of the 2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 8–10 November 2018. [[CrossRef](#)]
31. Huang, R.; Gu, J.; Sun, X.; Hou, Y.; Uddin, S. A Rapid Recognition Method for Electronic Components Based on the Improved Yolo-V3 Network. *Electronics* **2019**, *8*, 825. [[CrossRef](#)]
32. Shi, Z.; Xu, X. Near and Supersonic Target Tracking Algorithm Based on Adaptive Kalman Filter. In Proceedings of the 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 27–28 August 2016. [[CrossRef](#)]
33. Reza, Z.; Buehrer, R.M. *An Introduction to Kalman Filtering Implementation for Localization and Tracking Applications*; The Institute of Electrical and Electronics Engineers, Inc.: New York, NY, USA, 2018.
34. Liu, Y.; Wang, P.; Wang, H. Target Tracking Algorithm Based on Deep Learning and Multi-Video Monitoring. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018. [[CrossRef](#)]
35. Li, H.; Qin, J.; Xiang, X.; Pan, L.; Ma, W.; Xiong, N.N. An Efficient Image Matching Algorithm Based on Adaptive Threshold and Ransac. *IEEE Access* **2018**, *6*, 66963–66971. [[CrossRef](#)]
36. Tounsi, K.; Abdelkader, D.; Iqbal, A.; Sanjeevikumar, P.; Barkat, S. Extended Kalman Filter Based Sliding Mode Control of Parallel-Connected Two Five-Phase Pmsm Drive System. *Electronics* **2018**, *7*, 14.

37. Demirović, D.; Skejić, E.; Šerifović–Trbalić, A. Performance of Some Image Processing Algorithms in Tensorflow. In Proceedings of the 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, Slovenia, 20–22 June 2018. [[CrossRef](#)]
38. Beibei, Z.; Xiaoyu, W.; Lei, Y.; Yinghua, S.; Linglin, W. Automatic Detection of Books Based on Faster R-Cnn. In Proceedings of the 2016 Third International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC), Moscow, Russia, 6–8 July 2016. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).