A Deep Learning-Based Approach for Extraction of Positioning Feature Points in Lifting Holes

Jiahui Qian 1,†, Wenjun Xia 2, Zhangyan Zhao 1,*,† and Faju Qiu 3

1 School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China
2 China Railway Siyuan Survey and Design Group Co., Ltd., Wuhan 430063, China
3 Ningbo Special Equipment Inspection and Research Institute, Ningbo 315020, China
* Correspondence: zzy63277@163.com
† These authors contributed equally to this work.

Abstract: Due to uncontrollable influences of the manufacturing process and different construction environments, there are significant challenges to extracting accurate positioning points for the lifting holes in prefabricated beams. In this study, we propose a two-stage feature detection, which comprises the ADD (multi-Attention DASPP DeeplabV3+) model and the VLFGM (Voting mechanism line fitting based on Gaussian mixture model) method. Initially, the YoloV5s model is employed for image coarse localization to reduce the impacts of background noise, and the ADD model follows to segment the target region. Then, the multi-step ECA mechanism is introduced to the ADD. It can mitigate the loss of interest features in the pooling layer of the backbone as well as retain the details of the original features; DASPP is adopted to fuse features at different scales to enhance the correlation of features among channels. Finally, VLFGM is utilized to reduce the dependency of accuracy on segmentation results. The experimental results demonstrate that the proposed model achieves a mean intersection over union (mIoU) of 95.07%, with a 3.48% improvement and a mean pixel accuracy (mPA) of 99.16% on the validation set. The improved method reduces vertexes error by 30.00% (to 5.39 pixels) and centroid error by 28.93% (to 1.72 pixels), which exhibits superior stability and accuracy. This paper provides a reliable solution for visual positioning of prefabricated beams in complex environments.

Keywords: alignment feature points; DeepLabV3+; line fitting; visual lifting; YoloV5s

1. Introduction

Prefabricated components offer advantages such as a shorter construction time, lower costs, and better quality. As a result, prefabricated construction is gradually becoming the direction of development in the construction industry [1]. Prefabricated components are often transported through assembly and disassembly methods. However, traditional manual installation and dismantling methods for lifting equipment are unproductive and pose safety risks [2], which fail to meet the requirements in engineering. With the rise of smart construction, machine vision and image processing are being introduced into construction operations [3]. Non-contact approaches not only enable precise positioning of targets and improve the efficiency of many operations [4], but also reduce potential safety risks, providing greater opportunities for the sustainable development of prefabricated transportation operations.

Due to variations in scenes and operation methods, the strategies for visual alignment also differ, including robot-guided alignment [5], predicting component motion trajectories [6], and assessing assembly progress through landmark perception [7]. When evaluating the spatial orientation of prefabricated components, binocular vision (BV) is often employed [8,9]. This method utilizes feature matching points to obtain the object’s spatial information, such as spatial coordinates and angles, and adjusts the translation feed
of the motor based on the three-dimensional coordinates to achieve alignment assembly. However, using BV for alignment requires highly overlapping corresponding feature points and the acquisition of the target’s centroid position. Nevertheless, corner detection cannot provide centroid points, and it is more commonly used to solve the centroid through the target connected domain.

The prerequisite for obtaining the interest connected domain is to separate the hoisting hole area and background information in the image. However, the environment at the bridge lifting site is complex, and there may be factors such as lighting, shadows, and foreign matter within the target region. Traditional segmentation methods are easily influenced by these noise factors. DeepLabV3+ is a popular and effective pixel-level semantic segmentation model [10], even used to determine positioning points [11,12]. However, it has limitations in accurately identifying objects in large-scale scenes. To overcome this, researchers have introduced object recognition as a pre-processing step to minimize background interference. Deng [13] and Hsu [14] both adopted Yolo series models to address the segmentation of small objects. Additionally, morphological operations and geometric constraints are employed to refine the segmented images, resulting in more realistic representations of fine details. Furthermore, many studies use different methods to enhance the effect of edge segmentation, including using multi-level cascading residual modules to increase the range of the backbone receptive field [15], introducing the dense connection method in the ASPP module, reducing the spacing of feature sampling pixels [16], using improved convolution methods [17] and contour constraint methods [18].

Feature point detection methods for image salient features include edge-based corner detection methods and line feature-based detection methods [19]. However, these methods are often based on pixel gradients and are prone to noise points due to the influence of image texture. In order to meet the needs of different scenarios, Li [20] improved the CenterNet model to obtain target alignment feature points, but this method used for perspective view positioning can cause significant errors. Choi [21] adopted the FCN model to segment the target area, using the pixel with the highest probability of category within the segmented area as the matching point. However, the extraction accuracy of this method is greatly affected by the segmentation results. Guan [22] introduced a roundness threshold and obtained label information through the centroid, which significantly improved leak detection, but the results were affected by constraint conditions. In summary, existing studies generally rely on preprocessing results, it is impossible to accurately extract feature points without the effective preprocessing.

During the lifting operation of prefabricated beams, the extraction of feature points faces various challenges and difficulties due to factors such as the operating environment and manufacturing processes. In this paper, multiple approaches are employed to mitigate the impact of interference factors and the contributions are as follows:

- The lightweight YOLOv5s model is used for rough positioning of the lifting holes, reducing the impact of sample imbalance on pixel classification results.
- The ADD model is developed by improving the ResNet50 backbone network and introducing multi-stage attention mechanism and additional shallow channels. This model alleviates the influence of lighting conditions, ensures the integrity of complex image segmentation, and achieves higher accuracy.
- In order to mitigate the reliance on segmentation results, VLFGM is proposed to evaluate the coordinates of alignment points based on line features, and has better robustness and accuracy.

2. Methods
2.1. Overview

Considering the difficulty of directly obtaining corner points on the contours, firstly, target contour is obtained from original images. Then, VLFGM is utilized to determine the geometric shape of the rectangular hole, using the vertices of the fitted shape as feature points for image matching. Additionally, to achieve sufficient adjustment by utilizing an
adequate number of points, we calculate the centroid coordinates of the lifting holes, which hold significant importance and can even be applied on depth images. The workflow of this study is as follows (Figure 1):

![Flow chart of this study.](image)

**Figure 1.** Flow chart of this study.

### 2.2. The YoloV5s Model for Hole Detection

Although the Yolo series has been updated to version 8, YoloV5 [23] is still capable of handling many detection tasks. Additionally, in this scenario, the network model needs to be deployed on a hoisting device. In relation to existing models, we conducted several comparisons as shown in Table 1.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP0.50:0.95/</th>
<th>GFLOPs</th>
<th>Weight/M</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EfficientDet-b1</td>
<td>78.76</td>
<td>14.3</td>
<td>25.6</td>
<td>23.3</td>
</tr>
<tr>
<td>YoloV5s</td>
<td>87.60</td>
<td>15.9</td>
<td>17.1</td>
<td>54.9</td>
</tr>
<tr>
<td>YoloXs</td>
<td>87.72</td>
<td>26.8</td>
<td>34.3</td>
<td>43.1</td>
</tr>
<tr>
<td>YoloV7-tiny</td>
<td>86.24</td>
<td>13.2</td>
<td>14.4</td>
<td>91.0</td>
</tr>
<tr>
<td>YoloV8s</td>
<td>87.81</td>
<td>24.3</td>
<td>22.5</td>
<td>72.4</td>
</tr>
</tbody>
</table>

From Table 1, it is evident that YoloV5s outperforms EfficientDet in terms of both accuracy and FPS prediction. Despite the notable speed advantage of YoloV7-tiny, this model incurs a substantial graphics memory overhead during training, approximately 9.8 GB. In the context of the same dataset, training the YoloV7-tiny model demands three to four times the temporal investment required by the YoloV5s model, thereby implying escalated resource allocation for model training and refinement. While YoloXs marginally surpasses YoloV5s in mAP0.50:0.95, its heightened complexity engenders a processing lag of 11 frames in contrast to YoloV5s. Notably, YoloV8s fails to exhibit a significant ascension in accuracy. Furthermore, the YoloV8s boasts a larger parameter space, mandating augmented computational resources. Upon comprehensive juxtaposition of these outcomes, the YoloV5s model is elected as the preferred candidate, owing to its parsimonious parameter configuration and a judicious equilibrium between processing velocity and precision.

The YoloV5s feature extraction network consists of the Backbone, Neck, and Head components, as shown in Figure 2. First, the Mosaic data augmentation is applied within the Backbone, which can adaptively scale and crop the input images. The Focus module transforms high-resolution images into multiple low-resolution feature maps, capturing different-sized feature information. Additionally, the CSP (cross-stage partial network) structure enables local feature connections across stages, effectively reducing computation and parameters. The SPP (spatial pyramid pooling) structure captures semantic information at different scales, facilitating multi-scale fusion and improving the network’s ability to detect objects of different sizes.
The Neck layer consists of the FPN (feature pyramid network) and PAN (path aggregation network). The FPN layer utilizes bottom-up upsampling to propagate and fuse deep semantic features. Then, through top-down lateral connections, the bottom-level features are fused with the top-level features to form a multi-scale feature pyramid [24]. The PAN layer aggregates features layer by layer from shallow layers, and each feature map is fused with a high-resolution shallow feature map to preserve detailed features [25]. The Head layer predicts the location features and class information of the hoisting holes from the feature maps outputted by the Neck layer.

2.3. The ADD Network Model

2.3.1. Overall Structure

Despite being introduced in 2018, DeepLabV3+ [10] continues to exhibit the most favorable performance on the PASCAL VOC 2012 test dataset. The model demonstrates superior comprehensive capabilities and exceptional scalability. These attributes encompass the utilization of dilated convolutions to enlarge the receptive field, depthwise separable convolutions to reduce model parameters, multi-scale feature fusion for extracting semantic information from diverse layers, and the amalgamation of shallow and deep-level information to balance original and high-level features. Consequently, ADD is a semantic segmentation model based on the improved DeepLabV3+. The main optimization modules include: using three ECA modules to enhance the focus on the features of interest, replacing the ASPP (atrous spatial pyramid pooling) module with DASPP (dense connection atrous spatial pyramid pooling), and adding a shallow feature branch. The network structure of ADD is shown in Figure 3.

The proposed model still retains the encoder–decoder structure and uses ResNet50 as the backbone. In the encoding phase, the input image is first fed into the ResNet50-ECA residual network for feature extraction. The features outputted from the backbone network can be divided into two types: shallow semantic features of 256 channels from the 1st layer and deep features from the last layer. The shallow features are directly passed to the decoder, while the latter undergoes ECA (Efficient channel attention) weight assignment and further feature extraction through the DASPP module. The DASPP module includes parallel feature extraction operations, including 1 × 1 convolutions, dilated convolutions with dilation rates of 6, 12, and 18, and global average pooling, converting the 2048-channel backbone...
The proposed model still retains the encoder-decoder structure and uses ResNet50 as the backbone. In the encoding phase, the input image is first fed into the ResNet50 backbone network output into five 256-channel feature tensors. The DASPP module employs dense connections for each convolutional branch and these five differently processed feature tensors are concatenated to achieve multi-scale feature fusion. Finally, a $1 \times 1$ convolution is applied to convert the number of feature channels to 256, obtaining the high-level feature.

![Diagram](image)

**Figure 3.** The structure of ADD.

In the feature decoding phase, the input features consist of the original low-level feature 1, the ECA-processed low-level feature 2, and the high-level feature. The addition of the new low-level feature 2 has almost no impact on the network’s parameter count and provides better weight distribution for the shallow features, resulting in better preservation of the original information. Both shallow features undergo $1 \times 1$ convolution for feature dimension reduction. The deep semantic features are first upsampled by a factor of 4, reshaping the feature map to 1/4 of the original size. Then, the three feature tensors are fused together by addition, resulting in a 256-channel feature map. The fused features need further refinement, so a $3 \times 3$ convolution is applied to refine the feature map, changing its channel number and avoiding the occurrence of aliasing effects between the original features and the upsampled features [26,27]. Finally, the convolved image is upsampled by 1/4 to restore it to the original size and undergo pixel-wise classification to obtain the predicted image.

2.3.2. ResNet50-ECA

When selecting the backbone, we also considered using lightweight networks, similar to in [28]. However, it did not perform well on our dataset, and ResNet50 was chosen instead of MobileNet, as it has stronger feature extraction capabilities. ResNet50 [29] was proposed by introducing shortcut connections into the DCNN (Deep convolutional neural network) structure and using residual blocks (ResBlocks) as the main building blocks of ResNet. Each ResBlock contains two types of bottleneck structures, as illustrated in Figure 4. For a given input $x$, the output features of BottleNeck1 and BottleNeck2 are mapped to $x$ as follows:

$$H_1 = F(x, \{W\}) + Q(x, \{W_q\})$$  \hspace{1cm} (1)

$$H_2 = F(x, \{W\}) + x$$  \hspace{1cm} (2)
where \( F(x, \{ W \}) \) represents the feature maps obtained from the three convolution operations in the main branch, and \( \{ W \} \) denotes the 1 × 1 weight convolution operation. It is worth noting that for BottleNeck1, \( Q(x, \{ W \}) \) represents the feature maps from the branch, where \( \{ W \} \) also represents the 1 × 1 convolution operation.

![Figure 4](https://via.placeholder.com/150)

**Figure 4.** Two types of bottleneck. (a) Identity block. (b) Convolutional block.

Different bottleneck blocks serve different purposes, as shown in Figure 5a. The main function of BottleNeck1 is to alter the dimensionality and channel size of the feature map using a linear function, matching the feature map size with the input dimensions for subsequent operations. This enables the extraction of deeper features through multiple convolutional operations. On the other hand, BottleNeck2 is primarily used to maintain the input feature map’s dimensionality and channel size. It achieves this by employing a shortcut connection, directly adding the input “x” to the output “y”, which helps prevent the vanishing and exploding gradient problems during training deep networks. Additionally, in ResNet50, each ResBlock incorporates a varying number of shortcut connections, as indicated in Table 2. Since BottleNeck1 alters the shape of the feature map, it is generally not used in a sequential manner, whereas BottleNeck2, which preserves the tensor’s shape, can be concatenated.

![Figure 5](https://via.placeholder.com/150)

**Figure 5.** ResNet50-ECA module. (a) Improved ResNet50; (b) ECA mechanism.

<table>
<thead>
<tr>
<th>Module</th>
<th>BottleNeck1</th>
<th>BottleNeck2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResBlock 1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>ResBlock 2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>ResBlock 3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>ResBlock 4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

The conventional DCNN modules treat the semantic information of each channel equally important, which is not desirable when emphasizing specific features of interest. The ECA mechanism [30] addresses this issue by computing the global average value of the convolutional layer outputs to capture the channel-wise contributions, enabling channel feature compression. Subsequently, the channel weights obtained from the sigmoid
function are multiplied with the original outputs, enhancing useful features while suppressing irrelevant ones, as illustrated in Figure 5b. This approach facilitates the extraction of features of interest. ECA is a lightweight attention mechanism for convolutional neural networks. Unlike the traditional SE mechanism, it does not require computing the correlations between global positions, thus reducing computational and parameter overhead.

In Figure 5b, H, W, and C represent the height, width, and number of channels of the image, while X and X' denote the input and output, respectively. GAP stands for Global Average Pooling, and σ is the activation function that modulates and outputs the weight w for each channel. The optimal range for cross-channel information interaction, denoted as k, refers to the size of the 1-dimensional convolutional kernel. Its calculation is as

\[
k = \begin{cases} 
\left\lfloor \frac{\log_2(C) + b}{\gamma} \right\rfloor, & \text{if } \lfloor * \rfloor \text{ is odd}, \\
\left\lfloor \frac{\log_2(C) + b}{\gamma} \right\rfloor + 1, & \text{if } \lfloor * \rfloor \text{ is even}. 
\end{cases}
\]

(3)

where C is the number of channels, which is 2048 in ResNet50, \( \ast = \frac{\log_2(C) + b}{\gamma} \). Generally, \( \gamma \) and b are set to 2 and 1, respectively, and each parameter is substituted into the above formula to obtain a convolution kernel size of 7.

In the overall structure of ADD, as shown in Figure 3, the ECA mechanism is incorporated after the output of ResNet50 and at the low-level feature 2. At the location between layer 4 of ResNet50 and global average pooling, the inclusion of the ECA module enhances the network’s capability to express local features and expand the receptive field, thereby facilitating the capture of more diverse feature information. It is common for fully connected layer to compress feature tensors into lower dimensions, potentially resulting in information loss. By introducing the ECA module after the output of the fully connected layer, the information loss caused by feature compression can be alleviated (cascaded ECA mechanism). This approach can bring significant benefits, especially in tasks that require accurate modeling of global features.

Additionally, in the context of image semantic segmentation tasks, low-level feature maps play a crucial role as they contain the original details and texture information of the image [31]. These feature maps assist the model in identifying object boundaries and shapes. By applying the ECA module, adaptive channel weighting is performed on the feature maps, enhancing the correlation between each channel. This enables better exploration of fine details and texture information, leading to improved recognition of object boundaries and shapes. The concatenation of these three types of feature maps allows for the comprehensive utilization of their respective advantages, thereby enhancing the performance of the model.

2.3.3. DASPP

DeepLabv3+ employs the ASPP module, which utilizes cascaded dilated convolutions. By stacking dilated convolutions with different dilation rates, the output feature maps are concatenated together, gradually achieving a larger receptive field. The ASPP module addresses the trade-off between resolution and receptive field by employing dilated convolutions, enabling a larger receptive field size without increasing the number of convolutional parameters, thus encoding high-level semantic information. The calculation of the receptive field size R for the ASPP module is as follows:

\[
R = (r - 1) \times (k - 1) + k
\]

(4)

where r represents the dilation rate of the atrous convolution, and k indicates the size of the convolutional kernel. It can be observed from above equation that, while keeping other factors constant, R increases with the increment of the dilation rate. In DeepLabV3+, three
different dilation rates are employed with $3 \times 3$ convolutional kernels, specifically 6, 12, and 18. The corresponding receptive fields for each convolutional kernel are as follows:

$$\begin{align*}
R_{3,6} &= (6 - 1) \times (3 - 1) + 3 = 13 \\
R_{3,12} &= (12 - 1) \times (3 - 1) + 3 = 25 \\
R_{3,18} &= (18 - 1) \times (3 - 1) + 3 = 37 \\
\end{align*}$$  

By parallelizing 3 atrous convolution layers with different dilation rates, ASPP processes the same input feature and combines the results together, indicating that the output features are a multi-scale sampling of the input. To acquire a wider receptive field, a larger dilation rate is needed. However, as the dilation rate increases, the sampling rate becomes sparser compared to traditional convolution, resulting in the loss of more detailed information. At this point, the effectiveness of dilated convolutions diminishes.

The DASPP module addresses this issue by cascading multiple atrous convolution layers and densely connecting the output of each atrous convolution layer to subsequent ones. This cascading approach combines the advantages of both serial and parallel atrous convolution layers, enabling the generation of feature representations with more scales. As the dilation rate increases layer by layer, the output of each atrous convolution layer is combined with its input and the outputs of other layers, facilitating the fusion of multi-scale features and enlarging the receptive field [32].

The original dilated convolution layers are retained and cascaded in this study, combined with $1 \times 1$ convolutions to jointly encode and generate a denser feature pyramid. Furthermore, the concatenation of two dilated convolution layers enables the creation of a larger receptive field. Assuming that the receptive field sizes of the two dilated convolutions are $R_1$ and $R_2$, the resulting receptive field after concatenating these two dilated convolutions is illustrated in Figure 6.

![Figure 6. The impact of DASPP on receptive field.](image)

From Figure 6, it can be observed that the receptive field of two dilated convolution kernels with $r = 2$ and $k = 3$ becomes larger and denser. The specific calculation formula for the receptive field is as follows:

$$R = R_1 + R_2 - 1$$  

It should be noted that in DASPP, the $1 \times 1$ convolution does not affect the size of the receptive field. Additionally, the maximum receptive field for both ASPP and DASPP is:

$$\text{ASPP} : R_{\text{max}} = \max(R_{3,6}, R_{3,12}, R_{3,18}) = 37$$  

$$\text{DASPP} : R_{\text{max}} = R_{1,1} + R_{3,6} + R_{3,12} + R_{3,18} - 3 = 73$$
The comparison of maximum receptive field shows that the DASPP module used provides sufficient contextual information for pixel-level classification of the same image. This enables better segmentation details.

2.4. The Point Extraction Method Based on Line Features

2.4.1. Existing Problems and Solutions

Corner detection can extract feature points, but it is difficult to obtain the centroid points required for localization (as shown in Figure 7a). On the other hand, edge detection is susceptible to texture interference (Figure 7b), making the extraction of points of interest challenging. Pixel-level segmentation need to classify a large number of pixels, and due to the limitations of model performance, the image segmentation results may be biased towards larger or smaller regions in certain situations. In order to clearly demonstrate the influence of segmentation results on the centroid, we adopted an exaggerated approach to illustrate it, as shown in Figure 7c. When there are obvious misclassified pixel areas within the target region, it can result in missing vertices or centroid offset. Therefore, to reduce the influence of such special cases on the results, we abandoned the approach of extracting vertices based on edges or calculating centroids based on connected regions. This improvement can ignore boundary protrusions or depressions, resulting in less impact on the corner points and centroid calculation of the lifting hole.

![Figure 7](image_url)

**Figure 7.** Existing problems. (a) Harris corner detector; (b) canny edge detector; (c) the impact of segmentation on centroid.

2.4.2. Clustering Method for Boundary Points

Different situations call for different methods. If the target region is larger (Figure 7a), it is preferable to utilize the principal components of the boundary points for fitting. This approach allows us to disregard the protruding portion and only utilize the points within the normal range. Due to manufacturing processes and the influence of model segmentation performance, the boundary points of the segmentation results may exhibit situations depicted in Figures 7c and 8b, or even more severe cases. In such cases, fitting the convex hull points of the contour provides better results.

![Figure 8](image_url)

**Figure 8.** Fitting diagram. (a) Schematic diagram of judging CX; (b) fitting convex hull; (c) cluster results.

Determining the convexity (CX) or concavity (CA) of an image can typically be accomplished using the vector cross-product method, which yields the number of non-convex points and offers excellent scalability. The vector cross-product method involves calculating
the normal vector of a vertex by taking the cross-product of two adjacent edge vectors. Subsequently, the counterclockwise calculation of the normal vector for the polygon is performed. If the direction of the normal vector of a specific point is opposite to the direction of the polygon’s normal vector, the vertex is considered concave, as shown in Figure 8a. The specific calculation method is as follows:

1. For an N-sided shape, select any point \( P \) as the starting point, and counterclockwise construct an edge vector \( V_k \) from the endpoints of each edge, where \( k \leq N \).
2. For any three adjacent points \( P_m, P_t, P_n \), with their corresponding edge vectors denoted as \( V_m, V_n \), if \( V_m \times V_n \geq 0 \), \( P_t \) indicates a non-concave point; otherwise, it represents a concave point.

The discriminative illustration is shown in Figure 8a, where concave and convex points have normal vectors with different rotational directions.

Even if the predicted region appears to be a convex polygon, there may still exist some concave points in certain details. Therefore, we conducted tests on 200 predicted images and found that when the threshold for the number of concave points is set to 3, the algorithm can perfectly determine whether to use boundary points fitting or convex hull points fitting.

The boundary points are distributed in a strip shape in the two-dimensional space. For elongated samples with a flattened elliptical distribution (normal distribution ellipse), Gaussian mixture models are well suited for data clustering of this nature. The Gaussian mixture model (GMM) can be used to consider the sample points as a combination of multiple Gaussian distributions (GDs), where each GD corresponds to a cluster in the point set. Since GDs exhibit a bell-shaped curve in space, they can better adapt to the clustering of strip-shaped point sets. For a multivariate random variable following a GD, the expression is as follows:

\[
N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]
\]  

(9)

where \( \mu \) is the mean vector of dimension \( d \), \( x \in \mathbb{R}^{d \times 1} \) is the covariance matrix of dimension, and \( |\Sigma| \) represents the determinant of matrix \( \Sigma \).

In the GMM, each cluster is represented by a multivariate Gaussian distribution (GD), where \( p(x_i|\theta_j) = p(x_i|\mu_j, \Sigma_j) \). Let \( C \) represent the number of clusters. According to the above equation, each point \( x \) is a linear combination of \( C \) GDs. The purpose of the GMM is to estimate unknown parameters \( \Theta = \{w_1, \cdots, w_C; \mu_1, \cdots, \mu_C; \Sigma_1, \cdots, \Sigma_C\} \) based on the observed dataset \( X = \{x_1, \cdots, x_N\} \). For any observed value \( x_i \), let the set of parameters for the \( j \)-th cluster be denoted as \( \theta_j \in \Theta \), where \( 1 \leq j \leq C \). The probability of \( x_i \) belonging to the \( j \)-th cluster can be modeled as \( w_j|\sum w_j = 1 \), and the probability distribution of \( x_i \) can be expressed using the following equation:

\[
p(x_i|\Theta) = \sum_{j=1}^{C} w_j p(x_i|\theta_j)
\]  

(10)

The logarithm of the maximum likelihood function \( L(\Theta) \) can be used to evaluate the unknown parameters \( \Theta \), which can be expressed as:

\[
L(\Theta) = \log\ p(X|\Theta) = \sum_{i=1}^{N} \log p(x_i|\Theta) = \sum_{i=1}^{N} \log \sum_{j=1}^{C} w_j p(x_i|\theta_j) = \sum_{j=1}^{C} \log \sum_{i=1}^{N} w_j p(x_i|\mu_j, \Sigma_j)
\]  

(11)

\( L(\Theta) \) can be solved using the Expectation-Maximization (EM) algorithm [33], with the initialization of parameters \( w_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)} \). The recursive formulas for these parameters are as follows:

\[
w_{j}^{(s+1)} = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ij}^{(s)}, j = 1, \cdots, C
\]  

(12)
\[ \mu^{(s+1)}_j = \frac{\sum_{i=1}^{N} \gamma^{(s)}_{ij} x_i}{\sum_{i=1}^{N} \gamma^{(s)}_{ij}}, j = 1, \cdots, C \]  

(13)

\[ \Sigma^{(s+1)}_j = \frac{\sum_{i=1}^{N} \gamma^{(s)}_{ij} (x_i - \mu^{(s)}_j) (x_i - \mu^{(s)}_j)^T}{\sum_{i=1}^{N} \gamma^{(s)}_{ij}}, j = 1, \cdots, C \]  

(14)

The posterior probability \( \gamma^{(s)}_{ij} \) of given \( \Theta^{(s)} \) can be represented as:

\[ \gamma^{(s)}_{ij} = \frac{w^{(s)}_j N(x_i | \mu^{(s)}_j, \Sigma^{(s)}_j)}{\sum_{j=1}^{C} w^{(s)}_j N(x_i | \mu^{(s)}_j, \Sigma^{(s)}_j)}, i = 1, \cdots, N; j = 1, \cdots, C \]  

(15)

According to the assumption of independence, the next values of \( x, \mu, \Sigma \) need to be maximized by the expectation function, which is defined as:

\[ Y(\Theta^{(s)}) = \sum_{i=1}^{N} \sum_{j=1}^{C} \gamma^{(s)}_{ij} \log w_j N(x_i | \mu_j, \Sigma_j) \]  

(16)

Update the parameters in above equations sequentially until \( L(\Theta) \) is below a certain threshold. The performance of GMM heavily relies on the density of sample points. If the points are too sparse or too dense, it can lead to clustering errors. To address this, the distance between any two adjacent points was evaluated, and if the distance exceeded 12 pixels, an additional point will be inserted at the midpoint of those two points. This method effectively controlled the density of points. The resulting clustering outcome, as depicted in Figure 8c, clearly demonstrates the performance of this clustering approach.

2.4.3. Features Points Fitting

In certain cases, due to inadequate segmentation results, there may be some outliers within the given cluster groups, or even within the target region itself. The line fitting results are susceptible to the distribution of points, and to avoid the influence of such special cases, VLFGM is proposed. Its principle is as follows:

For a given set of points \( P = \{P_1, \cdots, P_n\} \), where two points \( P_i = (x_i, y_i), P_j = (x_j, y_j) \) form a line \( P_iP_j \), the equation of this line can be expressed as follows:

\[ y = \frac{y_j - y_i}{x_j - x_i + \epsilon} x + y_i - \frac{x_i(y_j - y_i)}{x_j - x_i + \epsilon} \]  

(17)

where \( \epsilon \) is a very small floating point number to prevent division by zero. Let \( A \) represent the slope of the line and \( B \) represent the intercept. The distance \( D(q) \) from any other point \( P_q = (x_q, y_q) \) in the set to the line \( L \) can be calculated as follows:

\[ D(q) = \frac{|Ax_q - y_q + B|}{\sqrt{A^2 + 1^2}} \]  

(18)

In order to directly and simply select the principal component points, we assign weights to each point in the point set based on their distance to the line. This approach considers points closer to the line as more important reference points, and the fitted line will be biased towards the intersection with points having higher weights due to their higher vote count. Specifically, we allocate scores to all points based on their distances,
such that closer points have a higher proportion of scores, while farther points have lower weights. This mapping relationship is detailed in Table 3.

Table 3. Mapping relationship between $D$ and $\psi$.

<table>
<thead>
<tr>
<th>$D$</th>
<th>$D = 0$</th>
<th>$0 &lt; D \leq 1$</th>
<th>$1 &lt; D \leq 2$</th>
<th>$2 &lt; D \leq 5$</th>
<th>$5 &lt; D \leq 10$</th>
<th>$D &gt; 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi(D)$</td>
<td>3</td>
<td>2.5</td>
<td>1.5</td>
<td>1.0</td>
<td>0.5</td>
<td>0</td>
</tr>
</tbody>
</table>

Therefore, the total vote count for the line $P_iP_j$ is indicated as

$$V = \sum_{q=1}^{n} \psi(D(q)), q \neq i, j$$

(19)

When a line receives the maximum number of votes, it indicates that it is the best fit line with the majority of points close to it. Thus, this line is considered as the optimal fitting line. Let us denote the starting point as $(x_0, y_0)$, and $n$ as the number of sides of the polygon. Naming all the vertices in a clockwise manner from 0 to $n - 1$. Once the four fitting lines are obtained, it is easy to determine the coordinates of the four vertices. For a given set of four vertices, which enclose a polygon, the centroid of the polygon can be quickly computed by:

$$Area = \frac{1}{2} \sum_{i=0}^{n-1} |(x_iy_{i+1} - x_{i+1}y_i)|$$

(20)

The coordinates of the centroid are given by:

$$x = \frac{\sum_{i=0}^{n-1} [(x_i + x_{i+1})(x_iy_{i+1} - x_{i+1}y_i)]}{6 \times Area}, y = \frac{\sum_{i=0}^{n-1} [(y_i + y_{i+1})(x_iy_{i+1} - x_{i+1}y_i)]}{6 \times Area}$$

(21)

3. Experiment and Model Training

Based on the actual position parameters at the site, a variety of experiments were designed in different environments. The visual effects of the dataset images are illustrated in the accompanying Figure 9. To ensure that the camera can capture the lifting holes completely, the camera setup at the site is shown in Figure 10b.

Figure 9. Different acquisition modes of images from the same perspective. (a) Normal illumination. (b) Color jittering. (c) Gaussian blur. (d) Low-light conditions. (e) Low-light color jittering. (f) Partial darkness. (g) Lighting shadows. (h) Image inversion.
Figure 10. Experimental equipment. (a) Simulation model; (b) experimental scenario.

To simulate a greater range of environmental variations, dataset creation process revolves around altering other conditions based on changes in perspective. When capturing images from specific angles, we introduce variations in lighting angles and intensity. Furthermore, for 20% of the original images, we apply 1 to 3 random augmentation techniques, including Gaussian blur, color jittering, and image rotation, to accommodate diverse scenarios. This approach ensures both the authenticity of the training set and the diversity of images within the dataset. Ultimately, after subjecting the initial set of 3102 images to partial augmentation processes, a total of 4083 images were obtained.

These images were randomly divided into training, testing, and validation sets in an 8:1:1 ratio. The training set was used for training the YOLOv5s model, and the cropped images of the holes were used as the training set for the ADD model. The software environment for model training was Windows 10 operating system with Python 3.7 and PyTorch 1.12. The hardware setup included an AMD Ryzen9 5900X CPU, an RTX 3080 (10 G) GPU, and DDR4 RAM (64 G). The camera model used for capture is HikVision MV-CH120-10GC, with a frame rate of 9.4 FPS and an image resolution of 4096 × 3000. The training process used the SGD optimizer with a batch size of 16 and 100 epochs. The learning rate was adjusted using cosine annealing, starting at 0.01. A momentum of 0.9 and a weight decay of 0.0004 were utilized.

To validate the effectiveness of the proposed method, it was compared with existing pixel-based classification methods, as well as DeepLabV3+-MobileNetV3. The evaluation criteria included model parameter size, mIoU, mPA, and segmentation result images. For pixel binary classification problems, mIoU and mPA scores are calculated using the following formulas:

\[
m\text{IoU} = \frac{\frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP} + \text{FN}}}{2} \tag{22}
\]

\[
m\text{PA} = \left( \frac{\text{TP}}{\text{TP} + \text{FP}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right)/2 \tag{23}
\]

where TP denotes the number of pixels correctly classified as positive class (Lifting holes), TN refers to the number of pixels correctly classified as negative class (Background), FP corresponds to the number of pixels incorrectly classified as positive class, and FN represents the number of pixels incorrectly classified as negative class.

Furthermore, in evaluating the fitting performance of the vertices, a comparison was made between the Hough line detector, line segmentation detection (LSD), LS, and VLFGM. In examining the centroid coordinate error, the proposed method was compared to the LS, minimum bounding rectangle method (MBR), edge centroid (EC), and connected domain centroid (CD). The aforementioned methods are compared based on the Euclidean distance error in terms of pixel points.
4. Analysis and Discussion of Experimental Results

4.1. Training Results of YoloV5s

Figure 11a presents the curves of training loss and mAP. It is evident that the curves show relatively smooth changes. Particularly, the mAP@0.5:0.95 indicator reached 87.60%, indicating high recall and precision of the model.

![Figure 11. Results of YoloV5s. (a) Training curves; (b) Detection results.](image)

Furthermore, as shown in Figure 11b. The left image demonstrates the detection results in the presence of various object interferences. The YoloV5s model accurately identifies the real lifting hole instead of adjacent interfering hole. The other three images depict environments with strong lighting, weak lighting, and lighting shadows, respectively. From the confidence of bounding boxes and the anchor boxes, it can be observed that the model is still capable of effortlessly handling target localization under these conditions.

Based on the experimental result images, YoloV5s consistently and accurately identifies the positions of lifting holes, regardless of different lighting conditions or shooting angles. This indicates that the model possesses excellent generalization performance and can reliably achieve rough positioning of lifting holes in the described lifting scenarios. These experimental results provide compelling evidence to demonstrate that YoloV5s can consistently acquire ROI and ensure the safety of lifting operations.

4.2. Image Segmentation Results

To evaluate the segmentation performance of the ADD model, this study compared its results with those of various methods, as shown in Table 4 and Figure 12.

**Table 4.** Evaluating indicators comparison of segmentation models.

<table>
<thead>
<tr>
<th>Models</th>
<th>mIoU/%</th>
<th>mPA/%</th>
<th>FPS</th>
<th>Parameter/M</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSPNet</td>
<td>91.74</td>
<td>97.22</td>
<td>20.76</td>
<td>49.13</td>
</tr>
<tr>
<td>SegNet</td>
<td>92.25</td>
<td>96.69</td>
<td>24.28</td>
<td>57.83</td>
</tr>
<tr>
<td>Unet</td>
<td>93.04</td>
<td>95.61</td>
<td>23.07</td>
<td>43.98</td>
</tr>
<tr>
<td>DeepLabV3+MobileNetV3</td>
<td>82.83</td>
<td>98.73</td>
<td>49.39</td>
<td>6.10</td>
</tr>
<tr>
<td>DeepLabV3+ResNet50</td>
<td>91.59</td>
<td>99.10</td>
<td>29.98</td>
<td>40.92</td>
</tr>
<tr>
<td>ADD</td>
<td>95.07</td>
<td>99.16</td>
<td>27.62</td>
<td>45.14</td>
</tr>
</tbody>
</table>

In Table 4, considering mIoU alone, DeepLabV3+ does not emerge as the best model. However, after considering the overall impact on the prediction results using mPA, we opted to improve this model. Unet, with its classic encoder–decoder structure, achieves the highest mIoU of 93.04% on the dataset. Nevertheless, its mPA is relatively low, indicating a higher number of misclassified pixels. As shown in Figure 12e, almost every predicted image from Unet exhibits missing boundaries, particularly in areas with strong lighting shadows. The unsophisticated upsampling and downsampling processes in Unet, coupled with the lack of global contextual information in the convolution operations, may result in the loss of features [34], causing the smaller predicted boundaries.
4.2. Image Segmentation Results
To evaluate the segmentation performance of the ADD model, this study compared its results with those of various methods, as shown in Figure 12c,d, when the segmentation results are relatively complete, PSPNet tends to produce less smooth edges, particularly at the vertices. The spatial pyramid pooling structure in PSPNet integrates features from different scales, but the varying semantic information across scales affects the perception of edges, leading to some features being suppressed and compromising edge smoothness. SegNet performs slightly better in terms of prediction results, but almost all images exhibit minor jaggedness along the edges. This is due to the lack of effective contextual information capture modules during the encoding phase, reducing the model’s segmentation capability for edge regions. When prominent lighting shadows are present, the aforementioned models mainly classify pixels in well-illuminated areas accurately.

Although PSPNet and SegNet have slightly lower mIoU scores compared to UNet, their higher mPA scores indicate more accurate prediction of region areas. However, both models have noticeable drawbacks. As depicted in Figure 12c,d, when the segmentation results are relatively complete, PSPNet tends to produce less smooth edges, particularly at the vertices. The spatial pyramid pooling structure in PSPNet integrates features from different scales, but the varying semantic information across scales affects the perception of edges, leading to some features being suppressed and compromising edge smoothness. SegNet performs slightly better in terms of prediction results, but almost all images exhibit minor jaggedness along the edges. This is due to the lack of effective contextual information capture modules during the encoding phase, reducing the model’s segmentation capability for edge regions. When prominent lighting shadows are present, the aforementioned models mainly classify pixels in well-illuminated areas accurately.

From Table 4 and Figure 12g, it is evident that the DeepLabV3+-ResNet50 model exhibits excellent overall performance metrics, segmentation results and FPS. This model achieves a remarkable mPA of 99.10%, indicating high accuracy in pixel classification and minimal missegmentation, making it well suited for segmenting hole images. However, DeepLabV3+ also has certain drawbacks. In certain images, similar to PSPNet’s segmentation results, distortion at the corner points can occur. Nevertheless, this phenomenon is somewhat alleviated due to the adoption of atrous convolution in DeepLabV3+, which offers a larger receptive field and richer semantic information. However, even with these improvements, discontinuities in connected regions still occur in images with strong lighting shadows. Additionally, we employed the lightweight MobileNetV3 as the backbone network. Unfortunately, despite boasting the best real-time performance, this approach yielded poor results across various aspects except for the high mPA, as depicted in Figure 12f.

The ADD model builds upon the DeepLabV3+ model by introducing the ECA mechanism and DASPP module. The improved algorithm incorporates attention mechanisms...
at multiple stages, resulting in more reasonable feature weight allocation and a better focus on relevant features. Moreover, the model’s parameter count increases only slightly, approximately 4.23 M. The densely connected multi-scale feature fusion module offers a larger receptive field and stronger semantic information extraction capability compared to the original model. As a result, ADD achieves the highest mIoU and mPA scores, as shown in Table 4. Simultaneously, the proposed model exhibits the best segmentation results across seven different environments, as illustrated in Figure 12h. In the first group of images, ADD’s predictions along the right edge do not exhibit downward bending, particularly noticeable in the lower right corner. Conversely, scenarios (d–g) manifest such curvatures, although these might seem present in the original images due to imaging effects.

In the second and fifth image groups, both ADD and scenarios (d–g) display jagged edges along the left and right boundaries. Within the fourth image group, ADD’s superiority over other models is distinctly pronounced. In the sixth experimental set, only ADD and the baseline show favorable outcomes. However, the baseline’s classification outcome in the upper left corner appears rounded and lacks sharpness, akin to an obtuse angle, which indicates evident distortion. Although some resulting images may exhibit unevenness, overall, the ADD model produces the smoothest edge effects. Furthermore, the model demonstrates superior resistance to lighting interference, preventing the disconnection of connected regions, which is crucial. The ADD provides a guarantee for subsequent detection tasks.

Moreover, we have conducted an analysis of the contributions of each module to the overall model, and the results are presented in the following Table 5. mIoU and mPA are employed as evaluation metrics.

Table 5. Evaluating indicators comparison of segmentation models.

<table>
<thead>
<tr>
<th>No.</th>
<th>Baseline</th>
<th>DASPP</th>
<th>Low Level ECA</th>
<th>Deep Level ECA</th>
<th>ResNet ECA</th>
<th>mIoU/%</th>
<th>mPA/%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>91.59</td>
<td>99.10</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>91.95</td>
<td>99.12</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>92.76</td>
<td>99.10</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>93.30</td>
<td>99.14</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>93.77</td>
<td>99.18</td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>94.06</td>
<td>99.12</td>
</tr>
<tr>
<td>7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>92.14</td>
<td>99.08</td>
</tr>
<tr>
<td>8</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>95.07</td>
<td>99.16</td>
</tr>
</tbody>
</table>

The DASPP connects features across different scales. As evident from Table 5, the introduction of DASPP leads to a 0.36% enhancement in the model’s mIoU metric. The incorporation of the ECA mechanism in low-level features results in a performance gain of 1.17%, signifying the retention of significant original information in the shallow network layers and emphasizing the favorable role of attending to these features. The introduction of attention mechanisms at the output end of the backbone network (referred to as the deep level in the table) optimizes both mIoU and mPA by 2.18% and 0.08%, respectively, underscoring the importance of deep-level semantic information for model performance, despite its susceptibility to loss and thus meriting profound consideration. Comparing experiments in groups 6 and 8, it is evident that introducing the ECA mechanism in ResNet effectively mitigates the impacts of operations such as pooling. Even with an already optimized foundation, the model’s mIoU metric can be further augmented by 1.01%, effectively counteracting excessive information loss caused by the backbone network’s pooling layers. By contrasting experiments in groups 4, 6, and 8, it is discernible that introducing attention mechanisms both before and after the backbone network’s output end can preserve essential feature information.
4.3. Analysis of Fitting Results

4.3.1. Visual Results Comparison

Through the aforementioned evaluation method, we conducted tests on point clusters under special circumstances. Whether the segmented region is oversized, incomplete, or even when the clustering results deviate, the VLFGM method consistently outperforms the LS, demonstrating more pronounced optimization effects on vertex coordinates. The comparative results are presented in Figure 13. By employing an improved approach to fit corner points, feature points from various segmented images can be accurately extracted. This technique also corrects deviations in vertex positions and compensates for the impact caused by incomplete image segmentation. The utilization of improved method enables more effective extraction and utilization of corner information in images, thereby enhancing accuracy and reliability in various application scenarios.

![Figure 13. Comparison of line fitting results. (a) Fitting CA by LS. (b) Fitting CX by LS. (c) Fitting CA by VLFGM. (d) Fitting CX by VLFGM.](image-url)

For images with better imaging quality, such as those in groups 1, 5, and 6 of Figure 12, fitting their key points for localization is not challenging. To clearly compare the fitting performance of different methods on the images with poor imaging performance, this study presents 5 groups of different result images, as shown in Figure 14. When the image segmentation results are relatively normal, the fitting results of Hough line detection, LSD, LS, and VLFGM are nearly identical and realistic, as depicted in Figure 14a,b. In this scenario, VLFGM degenerates into LS for obtaining higher scores in line fitting, considering the weighted votes of preferred points against other points. However, in special cases such as Figure 14c–e, there are some differences in the effects of different methods. Hough and LSD are feature-based methods relying on edge points or corner features, which are sensitive to significant gradient changes in pixels but cannot differentiate interest points, resulting in more noisy lines in the detection results and thus appearing more chaotic. Furthermore, LS cannot discard outliers in the points, leading to deviations in the fitted line slope as it directly regresses on the sample points. Consequently, LS fitting results exhibit larger errors in the vertexes of the enclosed shapes. In contrast, the proposed method can assign weights, extracting the principal component information from the sample data and outputting a line most related to the majority of points. This method demonstrates more stable fitting performance and, visually, can still achieve satisfactory results even in poorly performing images.

4.3.2. Error Analysis of Feature Points

A total of 60 experiments were designed to quantify the improvement effects of the proposed method, using GT as the baseline. The experimental subjects included both normal and abnormal segmented result images, and the comparative result is shown in Figure 15.
Figure 14. Comparison of line fitting results. (a,b) Normal shadow; (c-e) special cases.

Figure 15. Comparison of vertex errors. (a) Line chart of errors; (b) box plot of errors.

From Figure 14, it can be observed that the LS method is highly sensitive to abnormal images. Moreover, the fitting performance of this method heavily relies on the results of Gaussian mixture clustering, as depicted in Figure 13a,b. Due to these reasons, the LS did not perform well in the experiments, even though some points had fitting errors as low as 0 pixel, the overall error still reached 7.70 pixels, as shown in Figure 15. However, the VLFGM method demonstrates stronger anti-interference capability, as long as the principal components of the point clusters are normal, the fitting results will not vary significantly. Of course, when the segmentation results are extremely poor, the improved method may still exhibit larger error points, but compared to the LS method within the same set of experiments, it is relatively smaller. Overall, the VLFGM method concentrates the vertex
errors, and even when some image segmentation results are unsatisfactory, the average error in vertex fitting can still reach 5.39 pixels.

In order to prove from another perspective that the proposed method has better fitting accuracy, centroid errors were compared with other methods, and the experimental result is shown in Table 6.

Table 6. Comparison of centroid errors.

<table>
<thead>
<tr>
<th>No.</th>
<th>MBR</th>
<th>BP</th>
<th>CD</th>
<th>LS</th>
<th>VLFGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.81</td>
<td>2.99</td>
<td>4.25</td>
<td>2.47</td>
<td>2.14</td>
</tr>
<tr>
<td>2</td>
<td>1.60</td>
<td>1.48</td>
<td>1.03</td>
<td>2.02</td>
<td>1.12</td>
</tr>
<tr>
<td>3</td>
<td>2.92</td>
<td>9.39</td>
<td>4.79</td>
<td>2.76</td>
<td>2.46</td>
</tr>
<tr>
<td>4</td>
<td>3.26</td>
<td>1.85</td>
<td>2.06</td>
<td>1.25</td>
<td>1.03</td>
</tr>
<tr>
<td>5</td>
<td>0.75</td>
<td>0.75</td>
<td>1.18</td>
<td>0.56</td>
<td>0.56</td>
</tr>
<tr>
<td>6</td>
<td>4.07</td>
<td>4.12</td>
<td>2.96</td>
<td>2.30</td>
<td>1.68</td>
</tr>
<tr>
<td>7</td>
<td>1.41</td>
<td>1.84</td>
<td>1.97</td>
<td>1.77</td>
<td>1.50</td>
</tr>
<tr>
<td>8</td>
<td>3.75</td>
<td>5.48</td>
<td>3.86</td>
<td>5.25</td>
<td>1.35</td>
</tr>
<tr>
<td>9</td>
<td>2.70</td>
<td>2.25</td>
<td>1.35</td>
<td>1.95</td>
<td>1.25</td>
</tr>
<tr>
<td>10</td>
<td>5.02</td>
<td>4.35</td>
<td>7.22</td>
<td>3.88</td>
<td>4.07</td>
</tr>
</tbody>
</table>

Mean error 2.96 3.45 3.07 2.42 1.72

In theory, there is no direct relationship between vertex error and centroid error. Due to the presence of numerous internal points in the image, there can be error compensation phenomenon, even resulting in smaller centroid errors, as shown in Table 6. MBR exhibits larger centroid error, reaching 4.05 pixels, due to factors such as target pose during center extraction. EC relies solely on edge extraction for centroid estimation, which can lead to significant errors in cases of misclassification within the target region, as shown in Figure 12d. Therefore, EC performs slightly better than MBR. LS, by fitting quadrilaterals close to the original results, yields similar errors to CD. However, since the original connected regions contain more points, CD achieves a better average error performance, surpassing LS by 0.15 pixels. On the other hand, VLFGM is capable of fitting connected regions closer to the ground truth, resulting in a centroid error of only 1.72 pixels for the proposed method. This demonstrates its superior fitting performance.

5. Conclusions and Discussion

In this study, the YoloV5s model was employed to extract the lifting hole regions from panoramic photos. Subsequently, the ADD model was utilized for pixel-level segmentation of the target regions, followed by fitting the segmented areas using the VLFGM method to output the key point coordinates for alignment. Experimental results demonstrate that the proposed approach achieves higher accuracy in coordinate estimation, thereby providing assurance for spatial point extraction. The following conclusions are drawn from this research:

1. The YoloV5s model is capable of identifying lifting holes in various environments, with accurate boundary predictions of anchor boxes, ensuring reliable coarse localization and laying the foundation for subsequent tasks.
2. The ADD model exhibits superior stability and precision compared to other commonly used models, achieving mIoU and mPA metrics of 95.07% and 99.16%, respectively.
3. Even in cases where the segmentation results are suboptimal, the VLFGM method still performs well in most fitting tasks. It exhibits less dependence on image segmentation results compared to other methods. Moreover, the method achieves higher accuracy, with vertex coordinate errors reaching 5.39 pixels in the presence of abnormal images, reducing the overall error by 30.00%. The centroid error is as low as 1.72 pixels, representing a reduction of approximately 28.93%.
Of course, our method also has some limitations. The current weight assignment in the VLFGM method is predefined. However, weight acquisition in this method can also be enhanced through Gaussian distributions and machine learning, enabling better adaptive performance. Since we utilized Python programming, when the number of sample points is within 100, VLFGM might require approximately 0.05 s, involving the efficiency of the programming language and the code. Furthermore, construction operations prioritize detection speed and accuracy. Due to limitations in image resolution, and hardware capabilities, our method can only achieve a frame rate of 9.4 FPS. To enhance the real-time capability of the localization method, two approaches can be considered. First, increasing model prediction speed, including object detection and image segmentation, can be pursued by adopting a smaller backbone with a precise trade-off between accuracy and speed. Second, given the frame rate constraints of industrial cameras, employing faster frame rate image capture devices (which may involve lowering resolution, accuracy, or increased investment) constitutes another limitation of our approach. In practical operational scenarios, suspensions are often difficult to keep still. If noticeable vibrations occur, captured images by the device may exhibit blurring or other interferences, which is another aspect to consider.

Obtaining accurate coordinate information for key point extraction forms the foundation of visual alignment and is crucial for visual guidance. Future work will primarily focus on enhancing the usability of semantic segmentation models on target images, particularly improving the restoration of boundaries and vertices. Target region fitting can only serve as a mitigating approach. At the same time, visual localization has become an important direction in intelligent construction. More measures need to be taken to address the accuracy of feature point coordinates, including improving the real-time performance and accuracy of target detection and segmentation, reducing the complexity of coordinate point extraction, and ultimately achieving efficient and high-precision visual alignment to raise the level of intelligence in construction projects.

Author Contributions: Z.Z. put forward many constructive suggestions and provided experimental equipment; W.X., F.Q. provided some hardware devices, completed the experiment and recorded the data. J.Q. proposed specific ideas, solutions, and wrote this article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: In the process of completing this paper, we encountered numerous challenges, and I would like to express our sincere gratitude to Z.Z. for the guidance and to my colleague W.X. and F.Q. for their efforts in the experiments. Additionally, we would like to extend our heartfelt appreciation to the esteemed journal editors and the anonymous reviewers for their valuable assistance.

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

References
20. Li, Y. Robot grasp detection based on key point estimation. Aerosp. Syst. 2023, 6, 323–333. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.