DESIGN OF AN UNMANNED TRANSFER VEHICLE LOOP DETECTION SYSTEM FOR GRAIN DEPOT SCENARIOS
/
用于粮库场景的无人驾驶转运车回环检测系统设计

Boqiang ZHANG[1)], Dongding LI[1)], Tianzhi GAO*[1)], Kunpeng ZHANG[2)], Jinhao YAN[1)], Xuemeng XU[1)]
[1)] School of Mechanical and Electrical Engineering, Henan University of Technology, Zhengzhou 450001 / China;
[2)] College of Electrical Engineering, Henan University of Technology, Zhengzhou 450001 / China
Tel: 18003988576; E-mail: gtz1069312977@163.com
Corresponding author: Tianzhi GAO
DOI: https://doi.org/10.35633/inmateh-74-09

Keywords: *Grain depot, Food logistics, LCD, SLAM, Deep learning, Feature extraction*

ABSTRACT
The grain depot scenario is critical for grain logistics and transportation, and it is also a key setting for the efficient operation of intelligent grain logistics platform vehicles. A large number of repetitive and specific building structures, along with low-textured walls, characterize the grain depot scene. Loopback detection is an essential module in visual SLAM, and an efficient system can eliminate accumulated errors. While traditional systems rely on manually designed features, which struggle to adapt to the unique grain depot environment, this paper proposes a deep learning-based loopback detection system for grain transfer trucks. Leveraging a custom dataset capturing both grain depot environments and loopback scenarios, the system employs convolutional neural networks for identifying building equipment and door numbers, edge extraction for robust feature matching, and image template matching for efficient loopback verification. Extensive testing on the grain depot loopback dataset demonstrates that the system significantly improves loopback detection accuracy and efficiency, paving the way for reliable autonomous navigation in grain depots.

摘要
粮库场景是粮食物流转运的重要场景，同时也是智能粮食物流平台车高效运行的关键环节。粮库的大量重复和特殊建筑结构以及缺乏纹理的墙体颜色是粮库场景的特点。回环检测模块是视觉定位与建图的一个重要模块，有效的回环检测能够消除累积的误差，传统的回环检测使用的特征是人工设计的特征，在粮库的特殊场景下难以发挥出良好的效果。本文提出了一种基于深度学习的粮食转运车回环检测系统，利用录入了粮库环境和回环场景的定制化数据集，使用卷积神经网络识别建筑设备和门牌号码，通过边缘提取进行稳健的特征匹配，并采用图像模板匹配进行高效的回环验证。在粮库回环数据集上进行的广泛测试表明，该系统显著提高了回环检测的准确性和效率，为在粮库中实现可靠的自动导航铺平了道路。

INTRODUCTION
A nation's economy and quality of life are intrinsically linked to agricultural production, and food storage has been a cornerstone practice for farmers and traders throughout history. Grain storage is the main realization scenario of this paper, which is vital for the preservation of new grain, though significant breakthroughs in grain storage and transportation methods are still lacking. To tackle current problems in the grain storage and transfer process, an intelligent grain logistics platform vehicle has been developed to replace traditional grain transport trucks *(Zhang et al., 2023)*. This vehicle efficiently handles the transfer of grain from the raw grain cleaning center to various storage facilities. Compared to traditional large-scale grain transport trucks, the intelligent logistics platform vehicle offers features such as autonomous route planning, dynamic obstacle avoidance, and simultaneous localization and mapping in unknown areas. These capabilities effectively reduce long waiting times for numerous trucks during the harvest season in grain depot parks, health hazards to workers caused by harsh working environments, and traffic accidents within the parks. Vehicle navigation in these scenarios relies heavily on Simultaneous Localization and Mapping (SLAM) technology. Achieving accurate localization and map-building results is crucial for efficient operation. To balance effectiveness with cost, this research focuses on vision sensor-based SLAM.

Boqiang ZHANG, Senior engineer Ph.D. Eng.; Dongding LI, M.S. Stud. Eng; Tianzhi GAO, M.S. Stud. Eng; Kunpeng ZHANG, A.P. Ph.D. Eng.; Jinhao YAN, M.S. Stud. Eng; Xuemeng XU, Prof. Ph.D. Eng.

However, the unique features of grain depots pose challenges for traditional vision approaches. Their abundance of untextured granary buildings, floors, and specialized equipment can lead to unstable and inefficient feature detection and matching, particularly during loop detection.

Loop detection is a crucial part of the visual SLAM system, significantly reducing the cumulative error generated by the system by identifying and revisiting previously visited locations, thereby improving the accuracy of the constructed map. When the system detects a loop closure, it transmits this information to the backend for further optimization and error elimination, resulting in a more accurate map (*Gao et al., 2017*).

Appearance-based loop detection methods are indeed prevalent in visual SLAM (*Qu and Wang, 2011*), where rich visual information readily provides sufficient appearance cues for the system to rely solely on camera data, bypassing the compounding errors inherent in trajectory data. Consequently, in visual SLAM, loop detection essentially boils down to comparing image similarities.

Classical loop detection algorithms, as highlighted by Wu et al. and Qiu et al. (*Wu et al., 2010; Qiu et al., 2023*), often rely on manually designed features like SIFT, SURF, ORB, and BRIEF (*Rublee et al., 2011*) to represent images. However, this approach is not without its limitations. These traditional features, meticulously crafted by computer vision researchers, exhibit distinct characteristics: some are sensitive to environmental changes such as illumination variations, while others are hindered by computational complexity, limiting their broad applicability in diverse real-world scenarios.

With the rapid development of computer vision thanks to the continuous progress of deep learning techniques, CNNs have achieved great success in computer vision fields such as image classification, image segmentation, and target detection thanks to their powerful feature learning and representation capabilities (*Hongtao et al., 2016*). Since 2015, there have been attempts to use deep learning to extract features from images and thus replace hand-crafted features.

Xia et al. used the AlexNet network for feature extraction (*Xia et al., 2017*), followed by secondary training using Support Vector Machines (SVM) algorithm, and this loop detection model exhibited better robustness. Bai et al. proposed a CNN feature-based loop detection method that combines the pre-trained CNN intermediate layer output with the traditional sequence-based matching process output to reduce the computational complexity of the search strategy (*Bai et al., 2018*).

Mukherjee et al. used a deep deconvolutional network to represent the scene as a low-dimensional vector and determine the loop by comparing this vector (*Mukherjee et al., 2019*). Yang et al. proposed a parallel recurrent search and verification method that combines features from bag-of-words models and features from convolutional neural networks to act on loop detection (*Yang et al., 2021*).

Wang et al. used a two-stage loop detection strategy to avoid blind matching (*Wang et al., 2021*). Guo et al. used a VGG-19 network to extract the features of the images for the determination of loop detection by a locally sensitive hashing algorithm (*Guo Jizhi et al., 2021*). Scene-specific loop detection is still relatively rare.

To enhance loop detection efficiency in grain depot scenarios, this paper proposes a Convolutional Neural Network (CNN)-based approach for visual SLAM. This approach aims to improve both the accuracy and recall rate of loop detection. Grain depot environments are characterized by unique structures, such as towering silos, shallow round bins, and spherical bins. These structures are omnipresent in grain depots and pose challenges for traditional geometric feature-based methods. CNNs, on the other hand, offer long-term stability and robustness to perspective and illumination changes, making them ideal for feature extraction in these scenarios. Therefore, this paper proposes a deep learning feature-assisted visual SLAM framework specifically tailored for grain depot environments.

MATERIALS AND METHODS

In the grain depot, the shape of each depot is consistent and regularly arranged. The SLAM system is prone to classify the depots in different locations as the same scene when performing loopback determination, thus delivering wrong fitting information to the back-end and causing confusion in the system. In this paper, the system was divided into two branches. One is a lightweight GhostNet network, which extracts the deeper features of the image after transfer learning training. The other branch is a network for number recognition, which uses the grain depot door number to distinguish different grain depots, as shown in Figure 1.

The loopback detection system as a whole is shown in Figure 2.

Fig. 1 - Grain warehouse door number

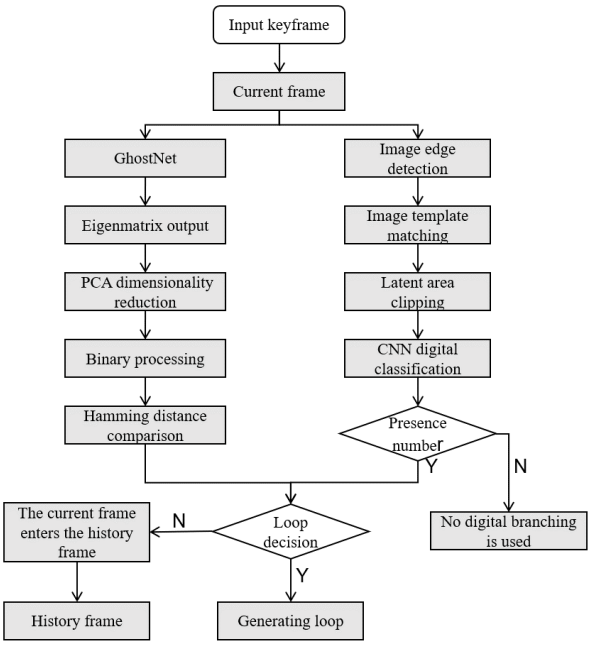

Fig. 2 - Algorithm flow chart

CNN Feature Extraction

CNNs have shown powerful performance within the field of computer vision for computer vision tasks such as target detection, image classification, and semantic segmentation. Traditional convolutional neural networks often contain a large number of parameters and complex computations, which are limited by the limited memory and computational resources of embedded devices, and it has become a new trend to study portable lightweight, and efficient convolutional neural networks (*BI et al., 2024; Feng et al., 2024*). For the above problems, the current common solution ideas are compact deep neural networks and efficient neural architecture design.

GhostNet network was proposed by Huawei Noah's Ark Lab in 2020, which is a lightweight CNN model with a smaller number of parameters and operations to ensure certain accuracy and can be deployed on removable embedded devices to meet the real-time requirements of visual SLAM systems. It divides the traditional convolution operation into two steps: the first step first generates feature maps with fewer channels using traditional convolution with less computation; the second step further generates more new feature maps using a small amount of computation on top of the generated feature maps using deep convolution; finally, the two feature maps are stitched together and the output is the final output. The idea of GhostNet is a phased convolutional computation module, which performs a linear convolution based on a few nonlinear convolutions to form a new feature map, and the large number of new feature maps obtained in this way is called Ghost of the previous feature maps. As shown in Figure 3, (a) figure shows the ordinary convolutional generation of feature maps, and (b) shows the Ghost module generation of feature maps.

The role of the GhostNet network itself is primarily to perform image classification and retrieval, rather than final output image features. From the network structure, the final fully connected layer serves as the output layer for image classification, and therefore is not considered as a feature extraction layer. The features extracted by the previous convolutional layer are too coarse to imply the global image. In this paper, the output of the FC8 layer of the GhostNet model is used as the features of the image, and the output of the FC8 layer is 1280 dimensional data.

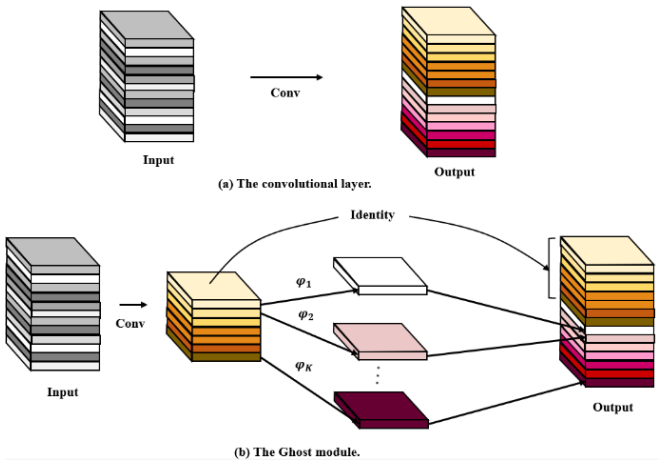

Fig. 3 - Comparison of normal convolution and Ghost module

Similarity Comparison

For the calculation of the distance between feature vectors, the more common and effective ones are the Euclidean distance and the cosine distance, if there are two feature vectors a[a1, a2, a3, ... , an]^T and b[b1, b2, b3, ... , bn]^T, then the cosine distance (*Zou and Umugwaneza, 2008*) between two vectors can then be expressed as:

$$d(a,b)=\frac{a^T\cdot b}{\sqrt{(a^T\cdot a)\times(b^T\cdot b)}} \tag{1}$$

When the feature vectors are of high dimension using the above two determination methods will be computationally intensive and affect the corresponding accuracy and precision, it is more important to use a more efficient computation method, which will help to improve the accuracy and real-time performance of the loop detection algorithm. Successful image retrieval methods have shown that data augmentation of the original feature vector can improve its ability to describe the image and increase computational efficiency. In this paper, Principal Component Analysis (PCA) (*Salih Hasan and Abdulazeez, 2021*) and binarization are used to augment the extracted features to improve the image feature representation.

The steps of data processing are as follows.

1) Calculate the covariance matrix Σ of the sample matrix, which is calculated as:

$$\sum=\frac{1}{m-1}X^TX \tag{2}$$

2) The SVD decomposition (Singular Value Decomposition) is performed on the covariance matrix Σ. The calculation is:

$$\sum=USV^T \tag{3}$$

The U and V unitary matrices in the formula are the left singular matrix and the right singular matrix.

3) The sample matrix is de-correlated using the left singular matrix U. The calculation is:

$$X_r=U^TX^T \tag{4}$$

4) Calculate the mean of each row of the sample matrix after dimensionality reduction.

$$mean=\frac{1}{m}\sum_{j=1}^{m}X_{ij},i=1,2,...,n \tag{5}$$

5) The sample matrix is binarized. The calculation is done as follows:

$$Y_{ij}\begin{cases}1,X_{ij}>=mean\\0,X_{ij}<mean\end{cases} \tag{6}$$

After the above process, the feature vector of each image is represented as a low-dimensional binary vector. The distance between two images can be expressed as the corresponding Hamming distance. The Hamming distance, which is the number of different elements of two equal-dimensional feature vectors at corresponding positions, is often used to determine the similarity between two images in the field of image retrieval.

Improved Canny Edge Extraction

Although the Canny edge extraction algorithm is easier to use and the edges are extracted more accurately, the traditional Canny algorithm has some disadvantages. Only Gaussian filtering is used in the image filtering stage, which is better for removing continuous noise such as Gaussian noise. However, Gaussian filtering generally uses pre-set conditions and is not able to take a more targeted filtering of the image based on the actual information of the image, which may make the image blurred. The image is unable to effectively filter out other types of noise, such as salt and pepper noise. In the final stage of the Canny edge extraction algorithm a human input threshold is used to determine the pixels in the image, resulting in poor adaptation to the image. In this paper, the traditional Canny algorithm is improved in two ways: the original Gaussian filter was replaced with a hybrid filter consisting of a Gaussian filter and an adaptive median filter, and an adaptive thresholding scheme was chosen to replace the fixed threshold in the final stage of the original algorithm.

The Gaussian filter can smooth the image and remove some low-frequency noise, and the adaptive median filter can remove the salt and pepper noise in the image, etc. The combination of the two can better remove the noise in the image, improve the quality of the image, and retain the details of the image.

Adaptive median filtering (*Yu et al., 2016*) is based on median filtering and addresses the window size problem, utilizing the advantages and disadvantages of filtering in both large and small windows, and adapting to change the size of the window according to the noise. After determining the filter window size, the adaptive median filter will set up judgment conditions to identify whether the median point is a noise point, which effectively avoids the problem of filter failure in median filtering. The adaptive median filter first constructs a rectangular window S with point (x,y) as the center point of the window. The following symbols are used to describe the principle: Z_{min} is the minimum gray value in the window S, Z_{max} is the maximum gray value in the window S, Z_{med} is the median gray value in the window S, Z_{xy} is the gray value of the coordinate (x,y) position, and S_{max} is the maximum window size allowed by S. The process of adaptive median filtering can be divided into two processes A and B.

$$A1 = Z_{med} - Z_{min} \tag{7}$$

$$A2 = Z_{med} - Z_{max}$$

$$B1 = Z_{xy} - Z_{min} \tag{8}$$

$$B2 = Z_{xy} - Z_{max}$$

If A1>0 and A2<0, go to process B. Conversely increase the size of the window. If after increasing the size of the window is not greater than S_{max}, repeat process A. Instead output Z_{med}. In process B if B1>0 and B2<0, then output Z_{xy}, and vice versa output Z_{med}.

It is inevitable that the edge information of the image will be lost when denoising with adaptive median filtering, in this paper, the determination of edge keeping was added in addition to the process of the determination of adaptive median filtering, and a new threshold was designed to protect the edge pixels. The calculation formula of the threshold value is (9).

$$T = \sqrt{\frac{1}{N\text{-}1} \sum_{i=1,j=1}^{N} (X_{ij} - \hat{X})^2} \tag{9}$$

In the formula, T is the threshold to be sought and $\hat{X}$ is the average value of the pixels in the window.

The gray values of the elements surrounding the center element in the window are used as the basis for determination. When the difference between the gray value of the center element and the gray value of the surrounding elements is greater than the threshold value T, the number of pixels accounting for one-fourth to three-fourths of the total number of surrounding elements, the center pixel is judged to be an edge point, and vice versa is judged to be a non-edge point. The image produced by adaptive median filtering and the edge information image are superimposed to complete the output. The pseudo-code for this part of the program is as follows:

Algorithm 1 Adaptive median filtering for edge preservation

$A1=Z_{med}-Z_{min}$, $A2=Z_{med}-Z_{max}$

If A1>0 and A2<0 do

$B1=Z_{xy}-Z_{min}$, $B2=Z_{xy}-Z_{max}$

If B1>0 and B2<0 do

Reserve Z_{xy}
Else do
Reserve Z_{med}
Else do
Enlarge window
If $S \leq S_{max}$ do
Return
Else do
Reserve Z_{xy}
$D=Z_{xy}-T$ (Take nine grids for example)
If $2<D<7$ do
Reserve Z_{xy}
Else do
Return
Merge image

Traditional Canny edge extraction algorithms use high and low thresholds to discriminate edge information, but the size of the high and low thresholds need to be set manually and have low adaptivity. In this paper, Otus adaptive thresholding algorithm (*Sha et al., 2016*) is used to give the high and low thresholds, the algorithm calculates the corresponding intra-class variance of the foreground and background through the different dividing values of the foreground and background parts of the detected target image, and the dividing value corresponding to the maximum value of the intra-class variance is the adaptive threshold calculated by Otus.

The coarse localization of the digital part uses the template matching technique in image processing technology. Template matching technique is a common image recognition technique, it first establishes a template library, the template library stores the content to be recognized, with the template library templates to traverse the input image, by searching for regions in the target image that match the given template to recognize a specific region in the image. In this paper, since only a coarse localization of the numbers on the granary is required, a template library consisting of numbers is constructed for template matching. In this paper, an image template library of 9 numbers from number 1 to number 9 is constructed with a size of 300pixel×300pixel, and an image can be selected from the template library to match with the input image when template matching is performed. The template library image is shown in Figure 4.

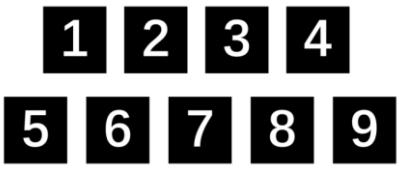

Fig. 4 - Digital template library

Construction of CNN for Number Classification

After coarse localization of the digital portion using image template matching, the portion where digits may be present is cropped from the image. The template matching technique achieves matching by finding the region in the target image that is most similar to the template image. However, this method may be affected by a number of factors that can lead to inaccurate matching. In contrast, convolutional neural network is a deep learning model that automatically extracts features from an image by learning a large amount of image data to achieve more accurate image recognition. Therefore, in this paper, a convolutional neural network is built to achieve accurate recognition of numbers.

The convolutional neural network built in this paper contains an input layer, a convolutional layer, a pooling layer, a fully connected layer and an output layer. The overall architecture of the network is shown in Table 1. The input to the network is a 28*28 image, a 5×5 convolution kernel is used in the first convolutional layer, followed by a 2×2 maximum pooling operation, and the above operation is repeated. At the end of the network is a fully connected layer and finally the network outputs the prediction through a fully connected layer of 10 neurons.

Table 1

Convolutional neural network structure

Operator	Input	Out	Kernel
Input	28 × 28 × 1	28 × 28 × 1	
Conv	28 × 28 × 1	24 × 24 × 32	5 × 5 × 1
ReLU	24 × 24 × 32	24 × 24 × 32	
MaxPool	24 × 24 × 32	12 × 12 × 32	
Conv	12 × 12 × 32	8 × 8 × 64	5 × 5 × 32
ReLU	8 × 8 × 64	8 × 8 × 64	
MaxPool	8 × 8 × 64	4 × 4 × 64	
FC	4 × 4 × 64	1024	
FC	1024	10	

Loop generation

The key to loop detection is to effectively detect the matter that a camera or other sensor device has passed through the same place (*Quan Meixiang et al., 2016*). If this thing can be successfully detected, more valid data can be provided to the back-end in a mature SLAM framework to get a globally consistent estimate (*Di et al., 2018*). Image selection needs to be taken into account when detecting image similarity; if the selection is too close, it will result in too much similarity between two frames, which will make it difficult to detect the frames inside the history frames that produce a loop (*Liu Guozhong and Hu Zhaozheng, 2017*). For example, the detection results in the nth frame being the most similar to the n-1 and n+1th frames, but obviously, such a loop judgment is meaningless. So, the order of the images should be processed in some way during the detection, assuming that the current frame is the nth frame and its neighbor has k frames of images, then the frame for loop similarity comparison should be outside the nth and kth frames.

RESULTS

Experimental environment

The equipment used in the test is an intelligent grain logistics platform vehicle, independently developed by Henan University of Technology. This self-driving vehicle is equipped with sensors such as LIDAR, a binocular camera, millimeter-wave radar, and RTK. It includes mounted devices like a storage unit, grain unloading system, wire control chassis, sensor module, and a core control unit computer. The vehicle can handle a 20-degree slope with a 3-ton load and features a large-capacity battery, ensuring it meets the operational requirements of a grain depot, as illustrated in Figure 5. The computer configuration used was an Inter(R) Xeon(R) CPU, an NVIDIA RTX3060 graphics card, 14 GB of RAM, and a software configuration of Python 3.8, CUDA 11.3, PyTorch 1.11.0, and an Ubuntu 20.04 operating system.

Grain Depot Environment Dataset and Model Training

For the grain depot scene in the actual environment, this paper uses monocular image acquisition equipment to produce a grain depot scene dataset, which has 1180 photos and divides the training set and test set according to 5:1. The dataset cover different environments with different lighting, shooting angles, shading, distance and size, which can easily reflect the existence of special buildings and special mechanical equipment in the grain depot environment.

The grain depot dataset produced in this paper contain three types of grain depots: cottage silos, shallow round silos, and vertical silos, which are widely used in most grain depots in China. In addition, the grain warehouse data set also includes common grain-related machinery and equipment for grain transportation, ventilation and drying, and bulk grain cleaning, such as steering conveyor, cleaning sieve, horizontal conveyor, mobile grain suction machine, bucket elevator, scraper conveyor, screw conveyor, grain picker, flat conveyor, tape conveyor, centrifugal ventilator, low-noise double-suction environmental protection centrifugal fan, grain warehouse insulation doors and windows, and weighing weighbridge. These devices play an important role in the harvest season, and these mechanical devices are generally used only in the grain depot environment and have a high degree of recognition. The grain depot data set is partially shown in Figure 6.

Fig. 5 - Henan University of Technology independently developed intelligent grain logistics platform vehicle

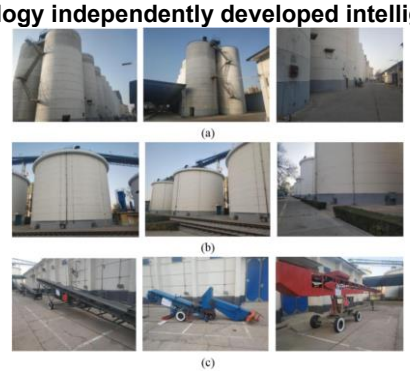

Fig. 6 - Grain depot dataset

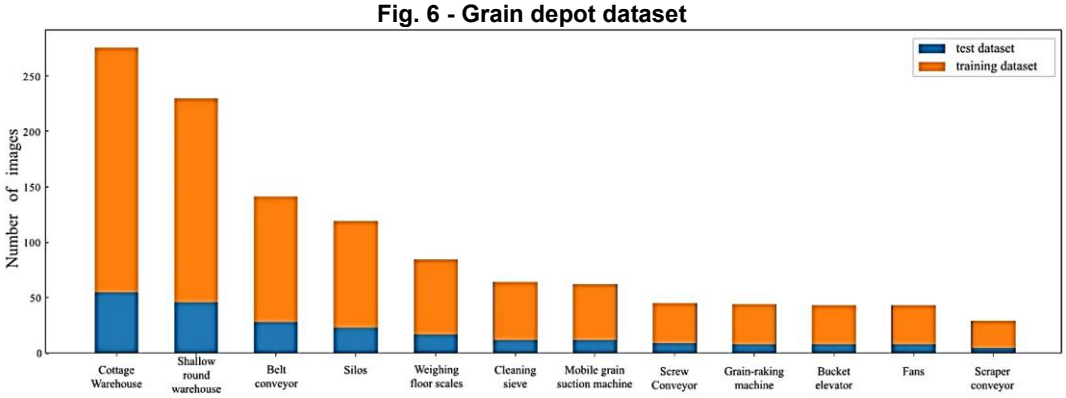

Fig. 7 - Number of images of each category in the grain depot dataset

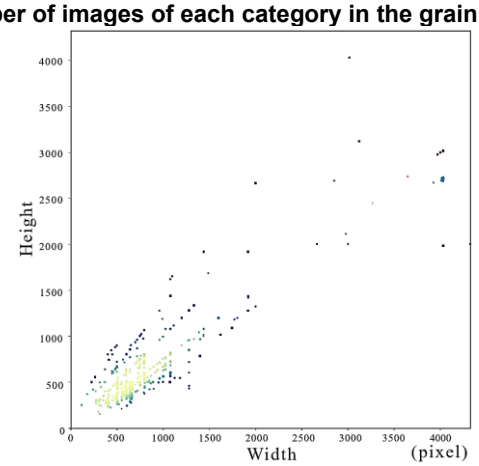

Fig. 8 - Distribution of grain depot dataset size

To address the limitation that the dataset produced in this paper cannot be as large as the world-renowned datasets, this paper employs data expansion strategies to augment the images within the grain depot dataset using various simple and effective methods, including flipping images left and right, random cropping, rotation, panning, noise perturbation, and luminance contrast transformation, thereby enhancing the model's robustness and adaptability to the grain depot scene. ImageNet dataset are computer vision dataset created by Fei-Fei Li, a professor at Stanford University, who led the creation of the ImageNet dataset. The dataset contains 14,197,122 images and uses pre-trained parameters to obtain good initial parameters for network training (*Deng et al.,2009*). This paper uses the GhostNet model trained on the ImageNet public dataset as the initialization weights for training the network, and this operation enables the network to show better performance in subsequent use.

SVHN (Street View House Number) Dateset is derived from Google Street View House Number and contains a large number of door numbers, as shown in Figure 9. The network is trained using the SVHN dataset as a way to adapt the classification of door numbers above the grain depot. In the grain depot environment, each depot has a consistent shape, and it is easy to classify depots in different locations as the same scene in the loopback detection system.

Fig. 9 - SVHN dataset

In this chapter training, Stochastic Gradient Descent (SGD) is used for training, the Famma of SGD is set to 0.1, the initial learning rate is 0.001, and 32 training images are selected for each iteration. Figure 10 and Figure 11 show the training of the pre-trained GhostNet network on the grain depot dataset. Figures 12 and 13 show the training of the SVHN dataset.

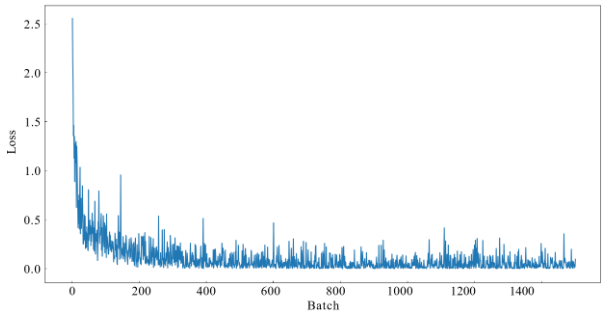

Fig. 10 - Loss function of training set

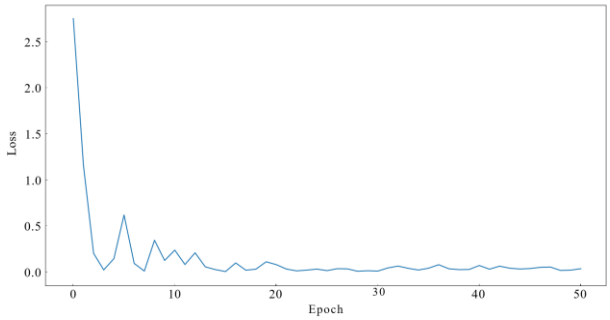

Fig. 11 - Test set loss function

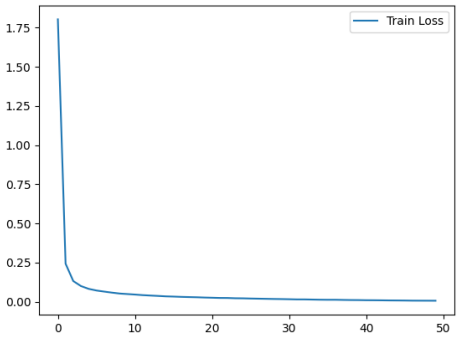

Fig. 12 - Loss function of training set

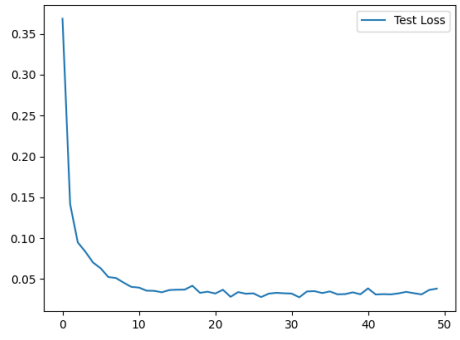

Fig. 13 - Loss function of training set

Edge extraction and template matching

The edge extraction algorithm is improved in the 'Improved Canny Edge Extraction' section of this paper, in which a hybrid filter combining Gaussian filter and adaptive median filter is used instead of the original Gaussian filter in the edge extraction algorithm, and adaptive high and low thresholds are used instead of manually setting the original high and low thresholds. This section makes a comparison between the hybrid filter and the improved edge extraction algorithm.

Gaussian noise and salt-and-pepper noise were added to the image, and Gaussian filter and the hybrid filter in this paper were used respectively for processing, and the effect was shown in Figure 14. From left to right, the original image, the image with Gaussian and Pepper noise added, the image processed using the Gaussian filter in the original algorithm, and the image processed using the hybrid filter in this paper, are shown.

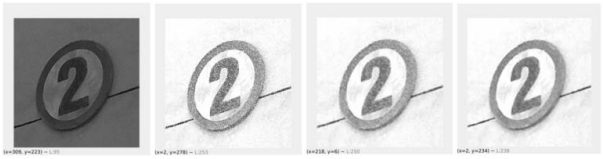

Fig. 14 - Comparison of the effect of filtering algorithms

The hybrid filtering algorithm proposed in this paper outperforms the Gaussian filtering algorithm used in the conventional Canny edge extraction method, both in terms of noise removal and edge information retention.

The traditional Canny edge extraction algorithm and the improved edge extraction algorithm are used to extract the edges of the grain depot image. The extraction results are shown in Figure 15.

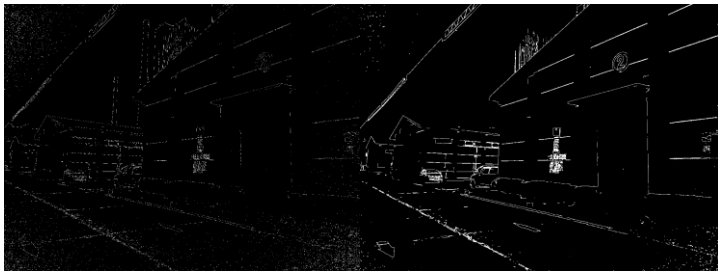

Fig. 15 - Comparison of edge extraction effect

It can be seen that the original Canny edge extraction algorithm is prone to extract more noise during edge extraction and the edges are not well protected. After improvement, it can show better performance.

The system performs edge extraction on the images in the template library as well as on the input image, followed by coarse localization of the target on the input image using a template matching algorithm. In this, the input images are processed using image pyramid, which reduces the amount of computation and time spent by utilizing images of different resolutions for multi-scale processing. At the same time, by using images of different scales for matching, the accuracy and robustness of matching can be improved. As shown in Figure 16. After coarse localization of the part of the input image that may be a digit the part is cropped to ensure that the digit occupies most of the area in the cropped image, and the cropped image is fed into the previously constructed convolutional neural network for accurate digit recognition.

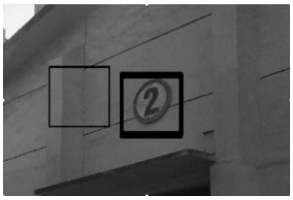

Fig. 16 - Results of image template matching

During the matching process, some parts that are not numbers can be matched, for example, the left box in Figure 16 is not a number, these parts can be well disposed of after entering into the convolutional neural network to ensure the accuracy of the system.

PR Curve Metrics

To verify the performance of the algorithm in this paper, comparisons are made in terms of Precision-Recall (PR) curve metrics, and extraction time of image features, respectively.

In the loop detection task, a classification can be made of the various phenomena that occur. The two images are judged by the algorithm to be the same scene as a loop. If the two images are not actually from the same scene, the phenomenon is called False Positive (FP); otherwise, it is True Positive (TP). If two images from the same scene are determined by the algorithm to be from different scenes, they are called False Negative (FN), otherwise, they become True Negative (TN). P and R in the PR curve are defined as shown in equation (10) (*Shin and Ho, 2018*).

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

Figure 17 shows the processing of the data after the features of the image have been extracted by the deep learning network. The performance when compared directly using cosine distance without data processing is different from the performance after performing principal component analysis to reduce the dimensionality and binarization. Therefore, in this paper, PCA and binarization were performed on the data.

The PR curve is an important metric for determining loop detection algorithms, and a good loop detection algorithm should have both high accuracy and recall. This paper uses the bag-of-words model DBoW3 and VGG16 (*Simonyan K. and Zisserman A., 2014*) for comparison, and VGG16 also uses PCA reduction and binarization for data processing.

Figure 18 shows the experimental results of the algorithm under the grain depot loop dataset, with the horizontal axis indicating the recall rate and the vertical axis indicating the correct rate. From the experimental results, it can be seen that the proposed loopback detection system in this paper has higher correctness and recall than the traditional bag-of-words model when performing loopback detection in a grain depot, and it is also more advantageous than a single convolutional neural network, which can be better applied to grain depot scenarios. When the classification of door numbers of grain depots is added, the accuracy and recall perform better than the single trained GhostNet network. Considering that there is not a door number in every location, the performance is only slightly better than a single network.

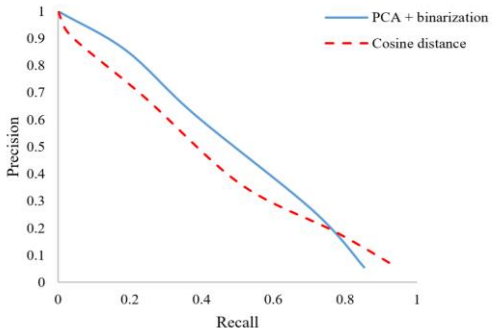

Fig. 17 - Data processing comparison chart

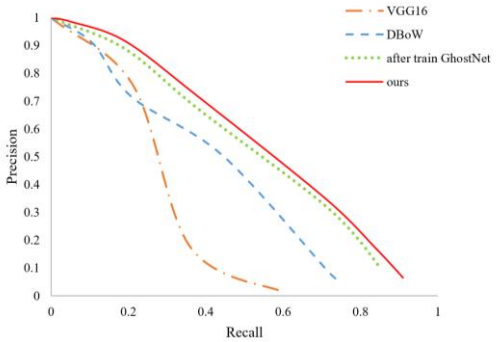

Fig. 18 - Grain depot scene loopback PR curve

CONCLUSIONS

Grain depots serve as critical lifelines for a nation's inhabitants. However, the influx of new grain each year poses numerous challenges that demand swift and decisive solutions. Employing modern equipment and technology in grain depots is a crucial path forward. This paper focused on addressing some of these challenges, specifically the repetitive building structures and low-textured environments that hinder the efficiency and accuracy of loop detection. Word bag models, commonly used in this context, suffer from limitations in both speed and accuracy.

This paper addressed the challenge of loopback detection in grain depots, where judging the similarity of unique buildings and consistent shapes can be difficult. A GhostNet architecture was leveraged to extract deep image features, which were then processed through PCA and binarization for enhanced representation. Additionally, a two-stage digit recognition branch was introduced. This branch utilized image template matching for coarse localization followed by CNNs for precise digit recognition. By combining these approaches, our loopback detection module for visual SLAM demonstrated robust performance in grain depots, paving the way for modernizing traditional grain storage facilities.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their constructive suggestions. This work was partially supported by the National Key Research and Development Program of China [Grant Number 2022YFD2100201], the Henan Provincial Key R&D and Promotion Project [Grant Number 231111241100].

REFERENCES

[1] Bai D., Wang C., Zhang B., Yi X., & Yang X. (2018). CNN Feature Boosted SeqSLAM for Real-Time Loop Closure Detection. *Chinese Journal of Electronics* 27(3): 488-499. https://doi.org/https://doi.org/10.1049/cje.2018.03.010

[2] Bi Z., Li Y., Guan J., & Zhang X. (2024). Real-time Wheat Detection Based On Lightweight Deep Learning Network Repyolo Model. *INMATEH - Agricultural Engineering.* 72: 601-610. https://doi.org/10.35633/inmateh-72-53

[3] Deng J., Dong W., Socher R., Li L. J., Li K., & Fei-Fei L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2009.5206848
[4] Kaichang D., Wenhui W., Hongying Z., Zhaoqin L., Runzhi W., & Feizhou Z. (2018). Progress and Applications of Visual SLAM.*Cehui Xuebao/Acta Geodaetica et Cartographica Sinica,* 47: 770-779. https://doi.org/10.11947/j.AGCS.2018.20170652
[5] Gao X., Zhang T., Liu Y., & Yan Q. (2017). 14 lectures on visual SLAM: from theory to practice. 206-234.
[6] Jizhi G., Fenglian L., Xinzhu Y., & Riwei W. (2021). The closed loop detection method of vision SLAM based on deep learning(基于深度学习的视觉 SLAM 闭环检测方法). *Journal of Optoelectronics Laser*. 32(06): 628-636. https://doi.org/10.16136/j.joel.2021.06.0392
[7] Hongtao L., & Qinchuan Z. (2016). Applications of deep convolutional neural network in computer vision. *Journal of Data Acquisition and Processing*. 31(1): 1-17.
[8] Guozhong L., & Zhaozheng H. (2017). Fast Loop Closure Detection Based on Holistic Feature from SURF and ORB (基于 SURF 和 ORB 全局特征的快速闭环检测). *Robot.* 39(01): 36-45. https://doi.org/10.13973/j.cnki.robot.2017.0036
[9] Mukherjee A., Chakraborty S., & Saha S. K. (2019). Detection of loop closure in SLAM: A DeconvNet based approach. *Applied Soft Computing*. 80: 650-656. https://doi.org/10.1016/j.asoc.2019.04.041
[10] Meixiang Q., Songhao P., & Guo L. (2016). An overview of visual SLAM (视觉 SLAM 综述). *CAAI Transactions on Intelligent Systems*. 11(06): 768-776.
[11] Qu, L., & Wang H. (2011). An overview of Robot SLAM problem. *2011 International Conference on Consumer Electronics, Communications and Networks (CECNet)*. https://doi.org/10.1109/CECNET.2011.5769022
[12] Rublee E., Rabaud V., Konolige K., & Bradski G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2011.6126544
[13] Hasan B. M. S., & Abdulazeez A. M. (2021). A Review of Principal Component Analysis Algorithm for Dimensionality Reduction. *Journal of Soft Computing and Data Mining* 2(1): 20-30.
[14] Sha C., Hou J., & Cui H. (2016). A robust 2D Otsu's thresholding method in image segmentation. *Journal of Visual Communication and Image Representation* 41: 339-351.
[15] Shin D. W., & Ho Y. S. (2018). Loop closure detection in simultaneous localization and mapping using learning based local patch descriptor. 30: 1-6.
[16] Simonyan K., & Zisserman A. (2014). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.2006.12567
[17] Wang Z., Peng Z., Guan Y., & Wu L. (2021). Two-Stage vSLAM Loop Closure Detection Based on Sequence Node Matching and Semi-Semantic Autoencoder. *Journal of Intelligent & Robotic Systems* 101(2): 1-21. https://doi.org/10.1007/s10846-020-01302-0
[18] Wu L., Hoi S. C., & Yu N. (2010). Semantics-Preserving Bag-of-Words Models and Applications. *IEEE Transactions on Image Processing* 19(7): 1908-1920. https://doi.org/10.1109/TIP.2010.2045169
[19] Xia Y., Li J., Qi L., Yu H., & Dong J. (2017). An Evaluation of Deep Learning in Loop Closure Detection for Visual SLAM. *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.18
[20] Yang Z., Pan Y., Deng L., Xie Y., & Huan R. (2021). PLSAV: Parallel loop searching and verifying for loop closure detection. *IET Intelligent Transport Systems* 15(5): 683-698. https://doi.org/https://doi.org/10.1049/itr2.12054
[21] Yu W., Ma Y., Zheng L., & Liu K. (2016). Research of Improved Adaptive Median Filter Algorithm. *Proceedings of the 2015 International Conference on Electrical and Information Technologies for Rail Transportation*, Berlin, Heidelberg, Springer Berlin Heidelberg
[22] Zhang B., Gao T., Chen Y., Jin X., Feng T., & Chen X. (2023). Research on unmanned transfer vehicle path planning for raw grain warehousing. *Journal of Intelligent & Fuzzy Systems* 45: 6513-6533. https://doi.org/10.3233/JIFS-232780
[23] Zou B. J., & Umugwaneza M. P. (2008). Shape-Based Trademark Retrieval Using Cosine Distance Method. *2008 Eighth International Conference on Intelligent Systems Design and Applications*. https://doi.org/10.1109/ISDA.2008.161