Scholar Publishing Group
International Journal of Neural Network
https://doi.org/10.38007/NN.2020.010303
ISSN 2790-2692 Vol. 1, Issue 3: 18-25

Deep Learning in Intelligent Navigation

Yonghu Hua*

Shenyang Institute of Science and Technology, Liaoning, China

**corresponding author*

Keywords: Deep Learning, YOLOv3 Model, Intelligent Navigation, Raspberry PI

Abstract: With the continuous development of computer vision, it becomes more and more important to make intelligent decisions through pure visual perception. Recently, deep reinforcement learning algorithms have been exploited to give intelligence to visual perception decision making. This paper mainly studies the application of deep learning in intelligent navigation. In this paper, YOLOv3 deep learning model is built and improved, the model is clipped, the deep learning model is lightweight, and the recognition speed is improved without significantly reducing the accuracy. Through the application and verification of the algorithm on the vehicle experimental platform, the real-time target tracking and autonomous navigation of the vehicle platform are realized.

1. Introduction

Autonomous navigation system is a hot problem to be solved in the field of robotics and autonomous driving. Satellite positioning and navigation technology is to obtain relatively accurate position coordinates, driving direction and moving path of moving objects by receiving information from GPS, Beidou and other satellite positioning and navigation systems [1-2]. Because the receiver of satellite positioning and navigation system is susceptible to signal interference, the positioning accuracy will fluctuate greatly in a certain working environment, resulting in inaccurate positioning or failure. Therefore, the satellite positioning and navigation system cannot independently cope with the real-time and stable navigation needs of robots and autonomous driving systems. In recent years, robotics and driverless technology have received wide attention and progress has been made in the field of Simultaneous Localization and Mapping (SLAM) [3]. SLAM was proposed to solve the problem of autonomous robot movement in an unfamiliar environment, allowing the robot to construct and record a map of the environment while autonomously moving. SLAM algorithm can be divided into two major parts: positioning and mapping. Positioning is to record the current position of the robot, which is represented by coordinates and directions, while mapping is completed by labeling obstacles or landmark objects [4]. The purpose of mapping is twofold: first, to enable the robot to complete navigation work in the recorded area; second, to provide localization assistance by correcting the map through loop feedback to reduce the positioning error of the robot. Specifically, SLAM is a method by which the robot can estimate the current coordinates while drawing the surrounding environment by collecting sensor information such as liDAR, millimeter-wave radar and camera [5]. Real-time positioning and map creation are important requirements for autonomous movement of robots and also the basis of robot intelligence.

Traditional navigation algorithms mainly include Simultaneous Localization and Mapping (SLAM) and path planning [6-7]. Compared with laser sensors, vision sensors have the advantages of low cost and retaining environment semantic information. In addition, researchers have proposed some visual SLAM algorithms, and some scholars have proposed Parallel Tracking and Mapping (PTAM) algorithm, which divides Tracking and Mapping into two independent tasks and processes, which are carried out in Parallel threads. One thread is tasked with tracking unstable handheld motions, while the other thread produces a point cloud of a 3D map through previously observed video frames [8]. An expert proposed ORB-SLAM to extend PTAM to more complex environments, and the feature extraction algorithm ORB has stronger feature extraction ability [9]. However, traditional visual SLAM has low robustness to the problems of fast camera motion, illumination transformation in the environment, strong rotation, texture images and so on[10].

The introduction of deep learning into the field of visual SLAM to solve various bottlenecks of traditional visual SLAM algorithms has become a hot research topic at home and abroad.

2. Construction of Intelligent Navigation Model based on Deep Learning

2.1. Principle of Deep Learning Algorithm

Neural network originated from the study of neurons in the field of biology, and the concept of artificial neurons was first introduced in perceptron. The basic principle of perceptron is that the input signal, after weighted and biased calculation, is processed by a nonlinear activation function to output the prediction result [11-12]. As shown in Figure 1, where x1... xn is the input signal, w1... wn is the weighted coefficient, B is the bias coefficient, Z is the result after the weighted sum, is the result after the activation of the nonlinear function, y is the predicted value.

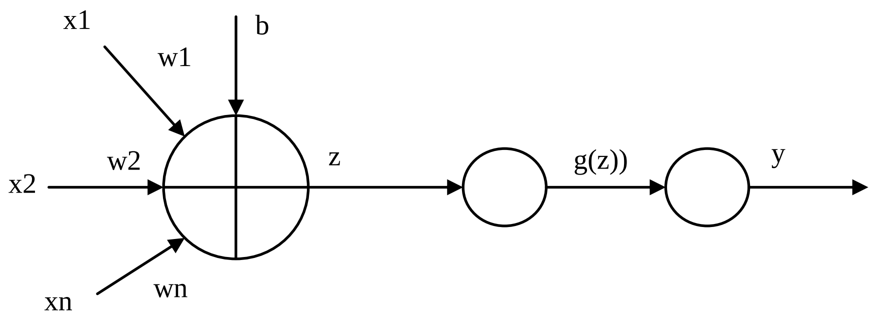

Figure 1. Schematic diagram of artificial neuron

The complex Network structure formed by the sequential linking of multiple Artificial neurons that can realize specific prediction function is called Artificial Neural Network (ANN) [13]. The core idea of artificial neural network is to set and organize a linear model with nonlinear activation functions (such as Sigmoid, TANh, Relu function, etc.) based on artificial neurons given input data and output form [14]. As for intermediate parameters, there is no need to set artificially. Output errors are evaluated by setting Loss Function, and Back Propagation (BP) is carried out through gradient descent and other methods to correct intermediate parameters, so as to minimize the Loss

Function [15]. When the loss function reaches the minimum, all parameters are obtained, and the training model is completed. Finally, the actual application data is brought into the model, and the prediction result can be fitted. General artificial neural network includes three parts: input layer, hidden layer and output layer, as shown in Figure 2.

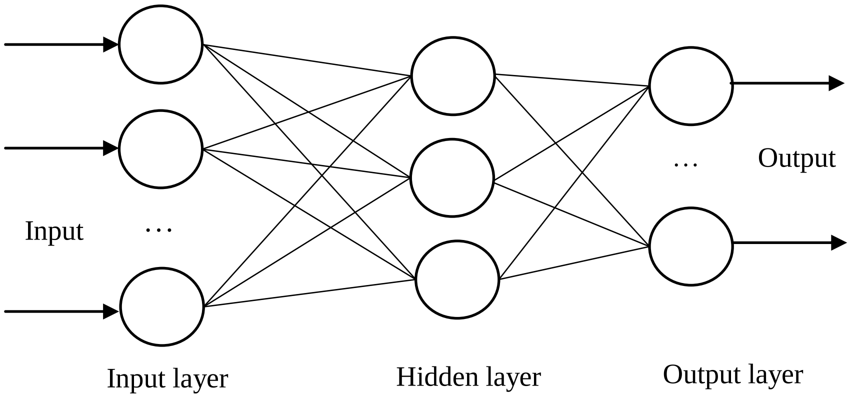

Figure 2. Schematic diagram of artificial neural network

However for images, every pixel in an image is the input parameter of a primitive, if the build neural network connects all the nodes and the input parameters to find characteristics, not only can cause too complicate network model to complex calculation, also will have serious over fitting phenomenon, make the network to learn too much irrelevant features and noise information, The generalization performance and stability of the model are reduced [16]. In order to solve this problem, people refer to the idea of Convolution kernel in traditional feature extraction, and extract image features by setting Convolution Layer to perform multiple Convolution operations on the image. Finally, these features are used to classify the objects in the image through the Full Connection Layer, and the results are output through the output Layer (Softmax). A Pooling Layer is set between the convolutional layers to perform operations similar to downsampling on images, which further accelerates the calculation and enriches the information of effective features [17-18].

In the convolution layer, the convolution kernels with a given size and number are used to calculate by successive window accumulation at the upper level. The parameters in each convolution kernel are equivalent to the weights in ordinary neural networks, which is more conducive to extracting local pixel features. The maximum pooling layer can be regarded as a special convolutional layer. The pooling operation does not output the result through the window calculation method of weighted sum, but adopts the maximum value in the output window as the output result. The calculation method of fully connected layer is the same as that of ordinary artificial neural network. Finally, the corresponding output model and loss function are designed according to different task requirements. Convolutional neural network has the characteristics of local feature perception and shared weight, which makes it more effective in image processing tasks. When an image contains multiple features, multiple features need to be obtained at the same time, and multiple convolution kernels of the same size can be used for extraction. The main method is to stack the convolution results of the input image and the output image together successively as the output, and the number of channels is equal to the number of convolution kernels used. The size calculation formula of the final output image is shown in (1) and (2).

$$M = \frac{m - f}{s} + 1 \tag{1}$$

$$N = \frac{N-f}{s} + 1 \tag{2}$$

Where M*N is the size of the output image, the size of the input image is M mu li, and is the size of the convolution kernel, and S is the convolution stride.

2.2. Network Structure Design

YOLOv3 is a convolutional neural network detection algorithm model based on no region suggestion. YOLOv3 adjusts the network model on the basis of previous YOLO series, and YOLOv3 adopts a larger scale and deeper network structure. With the increase of the depth of the network, although more feature information can be extracted, the training difficulty will be significantly increased. Therefore, in order to avoid gradient disappearance, YOLOv3 referred to the Resnet model, which finally improved the accuracy.

YOLOv3 introduced the idea of Anchor box in position prediction. The specific calculation method is shown in Formula (3), where Cx and Cy are the coordinate offset, Pw and Ph are the side length of the predicted target frame, and the final obtained (bx, by) is the central coordinates of the boundary frame, bw and bh are the width and height of the boundary frame, tx and ty are the central coordinates of the target frame, and tw and th are the learning objectives of the width and height of the boundary frame.

$$\begin{cases} b_x = \sigma(t_x) + C_x \\ b_y = \sigma(t_y) + C_y \\ b_w = P_w e^{t_w} \\ b_h = P_h e^{t_h} \end{cases} \tag{3}$$

Based on the YOLOv3 network model, this paper builds a lightweight network model that can run in the navigation system and identify specific obstacles in real time during the operation of the navigation system.

In order to achieve lightweight network, this paper mainly adopts the change of convolution mode and the reduction of layers. In this paper, depth-separable convolution is used as the convolutional layer to construct the network, because when the output channel is the same. The standard convolution operation will carry out multiple convolution operations, so in order to reduce the computation amount in the original network, the convolution layer is changed into separable convolution.

In this paper, based on the idea of YOLOv3 network structure, the convolutional layer extracts features, the pooling layer reduces parameters, and the network is reconstructed through the depth separable convolution layer and pooling layer. The convolutional layer is used together with the pooling layer to extract the features of the original image, and finally the number of layers in the whole network is reduced, so that the network is lightweight and the amount of computation is reduced, and the deep learning model can be run on embedded devices.

The main function of the residual network is to train the deep network and directly transfer the parameters to the deeper network layer. This paper trims the original network to reduce the number of layers of the network and reduce the calculation pressure of the deep learning model in actual operation, thus improving the identification speed. Moreover, this paper trims the trunk network, and the final network structure is more than twenty layers. The role of residual network in training is greatly reduced. YOLOv3 tiny is also one of the lightweight depth models. In this model, the residual layer is also eliminated, the network structure is very simple, and the speed of real-time detection is finally improved. Due to the small number of layers in the network, the improved network in this paper will not have the problem of gradient explosion or disappearance in training, so the residual network is not used and the network structure is simplified.

3. Implementation of Navigation System

3.1. System Hardware

The design of the hardware part of the system mainly includes the selection and integration of each function module. In this paper, Raspberry Pi 3B+ platform is selected as the basis of the control system, on which the steering gear and camera module are driven and the motor is used to drive and adjust the current output to achieve the purpose of motion control of the car, so as to ensure the normal running and realize the flexible acquisition of environmental information. At the same time, the wireless communication module is used to establish the communication with the upper computer to realize the real-time interaction of various information during driving. The system as a whole through the integration of the function of each module and cooperation operation, the formation of the data collection - transmission - processing - output, so as to achieve the automatic identification and tracking of the target object.

3.2. System Software

The software part of the system mainly includes startup and command input, through the parsing of commands to call different service functions, so as to achieve the corresponding functions. The main service functions of the system include camera image acquisition for information acquisition, data transmission and related computing processing, target identification and positioning, and motor output for controlling car movement. The image acquisition and motion control are equipped with the corresponding camera and motor drive module.

4. Autonomous Navigation Experiment

4.1. Identification Test

After the completion of the experimental platform, the target recognition algorithm needs to be tested. Firstly, the standard OTB data set was used to train the network on a PC, and the obtained model algorithm was transplanted to Raspberry pi 3B+ platform. At the same time, the same test picture is used to train and test on the PC side. Finally, the recognition efficiency and accuracy are compared. A total of 6 images from the test image database of Kaggle, an open platform, were selected as the test data.

Table 1. Platform validation test parameters

	1	2	3	4	5	6
Accuracy	91.7%	89.3%	88.4%	90.5%	87.1%	92.7%
Speed(s)	4	3	6	8	9	7

As shown in Table 1 and Figure 3, the algorithm running on both the experimental platform and the PC can accurately identify the target with an accuracy of more than 87%, but the recognition accuracy on the PC is slightly higher than that on the experimental platform, which is caused by the hardware performance advantage of the PC compared with the experimental platform. In terms of recognition speed, PC has more obvious advantages. Although image recognition can be carried out by running YOLOv3 model based on Raspberry pi 3B+ platform, the recognition speed is difficult for target tracking with demanding real-time requirements.

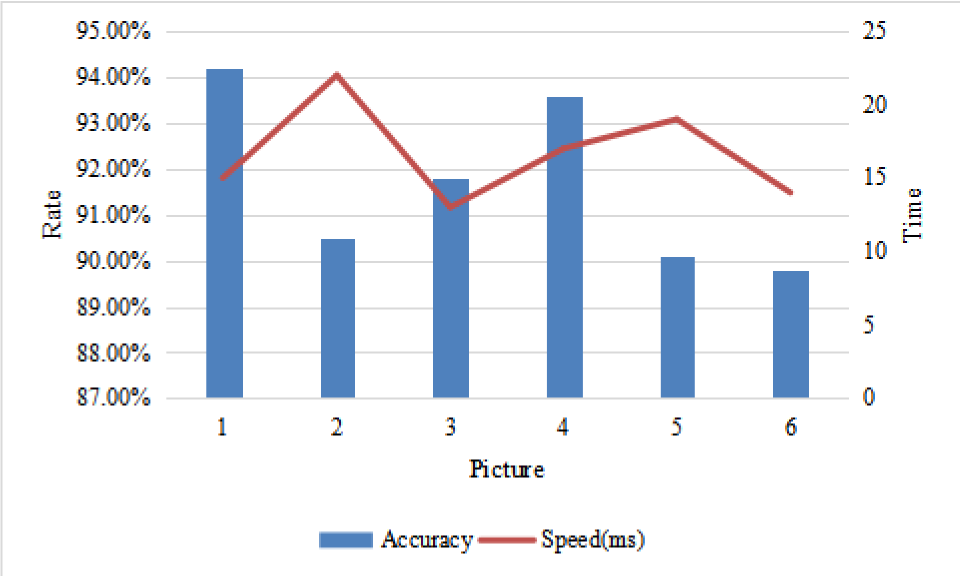

Figure 3. PC verification test parameters

4.2. Target Judgment

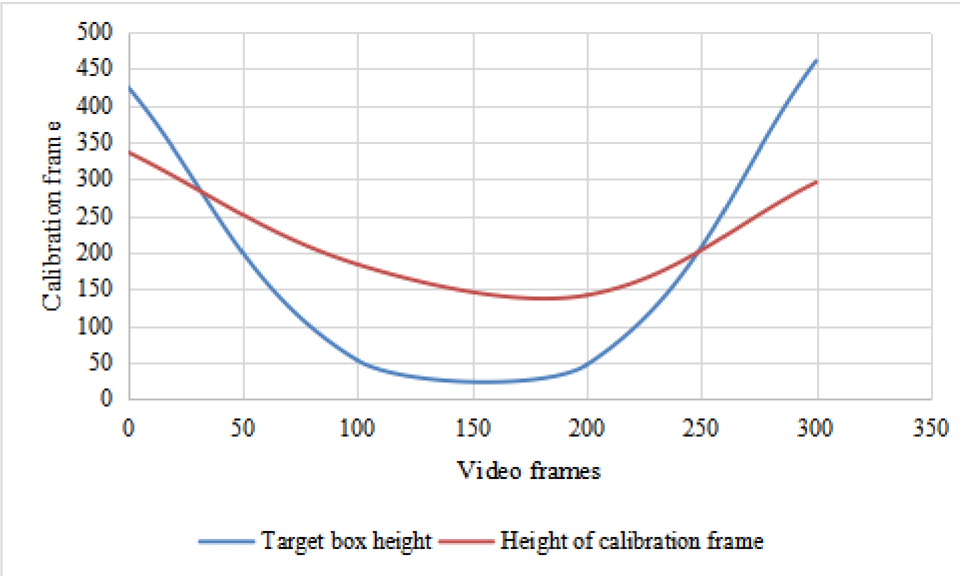

Figure 4. Change curve of width and height of object

As shown in Figure 4, the performance of the tracking and navigation control system of the smart car in this paper is evaluated by analyzing the specific position of the target car in the tracking video image. It can be concluded that the identification and calibration of the target object can always be kept within a certain range, indicating that the algorithm is accurate and effective as a whole. The smart car keeps the target in the field of vision by constantly correcting the relative position with the target, and keeps approaching the ideal position; At the same time, no matter what kind of movement the target car makes, the tracking car can make timely adjustment and keep the whole target in the image to maintain a certain size. In summary, through experimental verification and data analysis, it can be seen that the whole tracking and navigation system has good stability, accuracy, robustness and self-adjustment ability, and the intelligent navigation vehicle can achieve accurate real-time tracking of the target vehicle.

5. Conclusion

In this paper, through in-depth research and analysis of the environment perception, behavior decision, motion control and navigation auxiliary technologies related to intelligent vehicles and smart cars, the development and optimization of the core control algorithm, the design and development of embedded intelligent car tracking and navigation system as the core content, through simulation, build an experimental platform to verify the feasibility of the algorithm and the reliability of the function. The intelligent vehicle navigation system designed and developed in this paper provides a relatively simple and gentle environment for target tracking and global navigation, eliminating many uncertain factors and interference. However, the actual road driving environment is obviously more complex and changeable, full of uncertainty, which puts forward more stringent requirements for the performance of the control system.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Butaev M M, Babich M Y, Salnikovq I I, et al. Neural Network for Handwriting Recognition. Nexo Revista Cient fica, 2021, 33(2):623-637.

[2] Eckert A. Identification and Measurement of Hidden Costs in a Brazilian Turbine Manufacturer Company. Revista Gestão Industrial, 2018, 14(1):199-234.

[3] Bagwell J. Fuel Management: Key Areas To Watch. NAFA Fleet Executive, 2018, 11(6):9-9.

[4] Kosasi M H, Andi A, Arijanto L. Analisa Aspek-Aspek Dalam Prosedur Change Order Pada Proyek Konstruksi. Dimensi Utama Teknik Sipil, 2019, 6(1):9-16.

[5] Wiseman Y. Vehicle Identification by OCR, RFID and Bluetooth for Toll Roads. International Journal of Control and Automation, 2018, 11(9):67-76.

[6] Bulan O, Kozitsky V, Ramesh P, et al. Segmentation- and Annotation-Free License Plate Recognition With Deep Localization and Failure Identification. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(9):2351-2363.

[7] Obaidullah S M, Santosh K C, Das N, et al. Handwritten Indic Script Identification in

Multi-Script Document Images: A Survey. International Journal of Pattern Recognition & Artificial Intelligence, 2018, 32(10):1856012.1-1856012.26.

[8] Singh P K, Sarkar R, Das N, et al. Benchmark databases of handwritten Bangla-Roman and Devanagari-Roman mixed-script document images. Multimedia Tools and Applications, 2018, 77(7):8441-8473.

[9] Erdal, Erguzen, Atilla. Medical Image Archiving System Implementation with Lossless Region of Interest and Optical Character Recognition. Journal of Medical Imaging and Health Informatics, 2017, 7(6):1246-1252.

[10] Garg S, Kumar K, Prabhakar N, et al. Optical Character Recognition using Artificial Intelligence. International Journal of Computer Applications, 2018, 179(31):14-20.

[11] Ganesh S, Bhatti P T, Alkhalaf M, et al. Combining Optical Character Recognition with Paper ECG Digitization. IEEE Journal of Translational Engineering in Health and Medicine, 2021, PP(99):1-1.

[12] Rizvi S, Sagheer A, Adnan K, et al. Optical Character Recognition System for Nastalique Urdu-Like Script Languages Using Supervised Learning. International Journal of Pattern Recognition and Artificial Intelligence, 2019, 33(10):239-245.

[13] Konovalenko I A, Kokhan V V, Nikolaev D P. Optimal affine image normalization approach for optical character recognition. Computer Optics, 2021, 45(1):90-100.

[14] Mahesha C. Telugu Optical Character Recognition Using Cloud Computing And Python. International Journal of Computer Science Engineering and Information Technology Research, 2021, 11(1):95-106.

[15] Hubert, Phoenix P, Sudaryono R, et al. Classifying Promotion Images Using Optical Character Recognition and Nave Bayes Classifier. Procedia Computer Science, 2021, 179(1):498-506.

[16] Santoni M M, Chamidah N, Prasvita D S, et al. Penerjemahan Bahasa Indonesia ke Bahasa Minang dari Optical Character Recognition dengan Menggunakan Algoritme Edit Distance. Jurnal Ilmu Komputer dan Agri-Informatika, 2020, 7(2):105-113.

[17] Rajbongshi A, Islam M I, Rahman M M, et al. Bangla Optical Character Recognition and Text-to-Speech Conversion using Raspberry Pi. International Journal of Advanced Computer Science and Applications, 2020, 11(6):274-278.

[18] Rismanto R, Prasetyo A, Irawati D A. Optimalisasi Image Thresholding Pada Optical Character Recognition Pada Sistem Digitalisasi dan Pencarian Dokumen. Petir, 2020, 13(1):1-11.