Journal of Electronics and Information Science (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/jeis.2024.090310
ISSN 2371-9524 Vol. 9 Num. 3

A SLAM method based on deep learning

Hongjie Yu*

Energy-efficient Intelligent Systems Laboratory, University of Science and Technology of China, Suzhou, 215028, China
**Corresponding author: yhjf91@126.com*

Keywords: SLAM, deep learning, robust adjustments

Abstract: The core objective is to use deep learning to train an efficient feature detector, which provides a solid foundation for the construction of a feature point SLAM system. The training and optimization of deep learning models usually rely on large-scale labeled data, but for feature detection tasks, the annotation of feature points is abstract and subjective, which makes it difficult to obtain sufficient labeled image data. In order to overcome this challenge, this chapter proposes a dataset generation method that integrates traditional features combined with robust adjustments. By integrating two classical feature detection algorithms, we are able to generate fused feature points in natural scene images. These feature points not only combine the advantages of traditional features, but also enhance the generalization ability of the model through robustness adjustment. Based on this dataset, we use a deep learning framework for model training. By optimizing the network structure and loss function, we successfully trained a deep learning model that can detect two traditional feature points at the same time.

1. Introduction

In the first part, we performed traditional feature fusion on natural scene images to generate a dataset with key point labels. Firstly, different feature detection methods (such as SIFT, FAST, etc.) were used to extract feature points from the image and superimpose these feature points to form a preliminary feature point fusion. In order to solve the problem of congestion or overlap of key points that may exist in the preliminary fusion results, we designed a rule to calculate the response value of each key point, and used the non-maximum suppression (NMS) method to filter out the key points with higher local response values[1]. Then, by performing random Gaussian blur and Gaussian noise processing on the dataset, we obtained a high-quality image dataset with key point labels

Moving on to the second part, the joint training phase. Firstly, we use the Warp method to affine the label image to generate image pairs with key point annotations, and introduce photometric transformation to enhance the robustness of the model to photometric changes. In terms of network structure design, we combine MobileNet-related lightweight technologies and use the optimized VGGNet fully convolutional layer part as an encoder to extract features. In order to realize key point extraction and descriptive sub-computation at the same time, we use two lightweight and improved decoders[2]. Since the descriptors do not have direct label information, we use a dual-network architecture to calculate the descriptors separately and perform joint training. After

learning, we finally got a deep feature point detector that fuses traditional features.

Through this method, we successfully combined deep learning with traditional feature detection, improved the accuracy and robustness of feature detection, and laid a solid foundation for the subsequent feature point method SLAM system.

2. Feature fusion and adjustment

2.1. Feature extraction

In this study, two classical feature detectors, SIFT and FAST, were used based on the following considerations:

First of all, they perform well. Specifically, SIFT key points are highly regarded for their excellent stability; And the FAST key point stands out for its extremely fast extraction speed.

Secondly, SIFT and FAST have certain complementarity in feature detection. In a nutshell, SIFT mainly identifies the key points of the blob type, while FAST is better at capturing the key points of the corner type. This complementarity helps improve the accuracy and robustness of overall feature detection [3].

The descriptor is derived from the vectorization of the feature map. Therefore, when performing SIFT detection, we only need to extract the key points without the need to calculate additional descriptors [4]. This feature not only simplifies the inspection process, but also improves computational efficiency[5].

SIFT key points are used to detect and describe local features of the image, with scale and rotation invariance[6]. In this study, the OpenCV image processing library was used to quickly extract SIFT key points, including constructing a Gaussian-scale spatial pyramid to detect potential key points, locating and stabilizing key points through a fine model, assigning the dominant direction to ensure rotation invariance, and finally generating descriptors containing surrounding image information[7]. This process automates the extraction of key points and provides basic data for subsequent tasks.

Most of the coordinates of the SIFT and FAST keys extracted on the same image are inconsistent, indicating that they capture different image features [8]. SIFT focuses on the local maximum/minima region, while FAST focuses on corners or edges. This inconsistency highlights their complementarity, and when used together, it can provide a more comprehensive picture of image features and improve application performance, as shown in Figure 1

Figure 1: SIFT and FAST key point detection

2.2. Screening and fusion

Assuming that the key point set extracted by the SIFT method $K_s=\{p_i \mid i=1,...,m\}$, and the key point set extracted by the FAST method is $K_f=\{q_i \mid i=1,...,n\}$, where m and n are the number of

key points extracted by the two methods, and p_i or q_i are the pixel coordinates of the key points. In fact, when the two methods are used to extract the key points, the non-maximum inhibition method is used by default to prevent the key point bunching effect, but when the fusion is implemented, the pixel coordinates overlap or are adjacent, which can be described by equation (1).

$$\exists i,j,q\_i \in K\_s,p\_j \in K\_f \text{ such that } q\_i=p\_j \text{ or } \|q\_i-p\_j\|\_2\hat{}2\leq 2 \quad (1)$$

To avoid redundancy of this data, you need to design rules to remove one of them. For overlapping points, simply delete one of them at random; For adjacent points, the criterion of the maximum local gray difference is used to filter: in the sliding window of 3×3, the root mean square difference between each key point and the gray value of other pixels in the sliding window is calculated, and it is used as the response value of the key, and the NMS method is used to retain the key point with the largest response value in the region and delete the other key points.

As shown in Figure 2, the key points in each image after the fusion of features are screened to ensure that there is no coincidence or adjacency in each image.

Figure 2: Comparison before and after key point screening

2.3. Blur and noise

After the detection and fusion steps of the key points, we completed the image dataset with the key point labels. In order to ensure the robustness of the subsequent training results, the dataset needs to be adjusted to meet the common challenges of viewing angle, lighting variation, image blur, and noise in SLAM[9]. In view of the robustness of the change of perspective, the Warp operation has been embedded in the network structure, and the homology transformation is used. In addition, adjustments such as blurring, noise, and photometric transformations are required to make adjustments to the image, which are considered data augmentation in deep learning. Blurring and noise processing are performed before data is entered into the network[10]. The Gaussian ambiguity is achieved with a low-pass filter, and the larger the kernel size, the higher the degree of ambiguity. A high-pass filter is used to add Gaussian noise and the mean of the kernel function is adjusted to control the noise intensity.

2.4. Photometric transformation

In the process of joint training, the photometric transformation of the affine transform image is further implemented, and the gamma adjustment of the image is performed by using the luminance nonlinear point-by-point adjustment, and the image brightness gamma adjustment is shown in equation (2).

$$I'_{(x,y)} = \left(I_{(x,y)}/255\right)^{\gamma} \cdot 255, 0.5 \leq \gamma \leq 2 \quad (2)$$

Among them, I_((x,y)) and I_((x,y))^' are the gray values and correction values of pixels (x,y), respectively, and are γ adjustment parameters, with a value range of [0.5,2]. When the γ value is less than 1, the image brightness will be increased, and vice versa, the image brightness will be decreased.

In natural scenes, changes in lighting have a significant impact on image detail. This change can affect the performance of the detection and descriptors of key points in the image. In order to enhance the robustness of the network to lighting changes, we included image pairs under different lighting conditions into the training dataset.

By simulating different lighting conditions, we were able to train the network to learn and adapt to differences in image detail caused by lighting changes. This training method helps to improve the confidence of the detected feature points and optimize the performance of the feature descriptor. Therefore, in practical applications, even in the face of complex and changeable lighting environments, the network can more accurately and reliably identify and extract key features.

2.5. Key point detection and descriptor sub-decoding

In view of the process of detecting key points first and then calculating descriptors in the traditional feature point detection task, a new feature point detection model called MVP (Mobile-VGGNet-Point) was designed. The model integrates the lightweight design skills of MobileNet and the VGGNet-style fully convolutional neural network to improve the efficiency and accuracy of feature point detection. At the heart of the MVP model is the design of its shared encoders. As shown in Figure 3, the input image first passes through this shared encoder for the necessary processing and dimensionality reduction. The data after the encoder is offloaded to two separate paths: the key-detection decoder and the key-descriptor subcomputational encoder. The two paths are each responsible for different tasks, but most of the network parameters are shared, enhancing the ability to learn and share between the two tasks.

The key detection decoder is responsible for pinpointing the location of feature points from the encoded data. The key descriptor computational encoder focuses on generating a unique and stable descriptor for each detected key point. This design enables the MVP model to complete the position detection of feature points and the calculation of descriptors at the same time in a single forward propagation, which greatly improves the computational efficiency.

In addition, the full-scale image processing capabilities of the MVP model allow it to cope with inputs of various scales and resolutions, enhancing the versatility and usefulness of the model. Combined with the lightweight design of MobileNet, the MVP model reduces the computational complexity and model size while maintaining performance, making it more suitable for running on resource-constrained devices.

The encoder of the MVP network architecture was originally a VGGNet-style fully convolutional neural network with eight 3×3 convolutional layers. In order to improve efficiency and performance, this study uses the lightweight technology of MobileNet to improve. Improvements include: replacing standard convolutions with depth-separable convolutions to reduce the amount of computation and parameters; The inverted residual structure is introduced to improve the network representation ability by increasing the number of channels and then decreasing. The residual connection is used to enhance the learning ability of the network. These improvements make the MVP encoder lightweight while maintaining high performance, making it more suitable for running on resource-constrained devices.

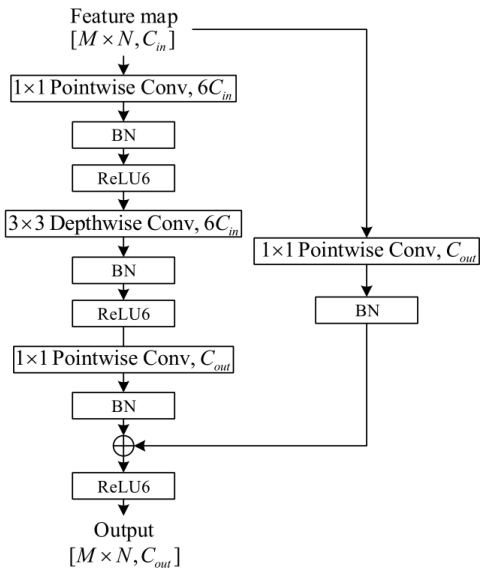

Figure 3: Encoder convolution block structure diagram

The network structure also includes a pooled spatial downsampling structure, which is connected to a maximum pool layer of 2×2 after every two layers, which is used to reduce the dimensionality of the original image under the premise of retaining features. A batch normalization optimization technique was also used, using ReLU6 (the slope of the first quadrant curve is 6) as the nonlinear activation unit. Assuming that the size of the input image is H×W, the feature map with the size of H_c×W_c is obtained through the three maximum pooling layers, where H_c=H/8 and W_c=W/8. From a local point of view, the 8×8 pixel grid area of the original image is obtained by the action of three non-overlapping ×2 maximum pooling operations of 2×2 in the encoder. In general, the encoder maps the input image I∈R^(H×W) to an intermediate tensor B∈R^(H_c×W_c×64) with a smaller space size and a larger channel depth.

3. Comparison and analysis of results

In this study, the learned MVP feature detector was used to conduct experiments on the illumination and viewing angle change images of the Hpatches dataset. The MVP detector integrates key point detection and descriptor sub-computation. In order to evaluate its reproducibility in different scenes, the non-maximum suppression (NMS) method was used to screen the key points, and the NMS suppression distance was set to 4 and 8 in the lighting and viewing angle changes, respectively. According to equation (2), the MVP model achieved a repeatability score (=3) for the key point detection part in both scenarios.

Table 1: Comparison of the repeatability of corner detection.

Key Point Detector	Lighting change scenes		Perspective change scene	
	NMS=4	NMS=8	NMS=4	NMS=8
MVP(ours)	0.657	0.623	0.512	0.487
SuperPoint	0.652	0.631	0.503	0.484
MagicPoint	0.575	0.507	0.322	0.260
FAST	0.575	0.472	0.503	0.404
Harris	0.620	0.533	0.556	0.461
Shi	0.620	0.511	0.552	0.453

As shown in Table 1, among the four evaluation data, the MVP model proposed in this study

performed better in two key indicators, namely the NMS rejection distance of 4 in the lighting change scene and the NMS rejection distance of 8 in the viewing angle change scene. Compared with the SuperPoint method, the MVP model has improved scores in most cases, only slightly worse in the case of lighting changes and the NMS rejection distance of 8. Overall, the MVP model has the best performance in corner detection repeatability, which verifies the effectiveness of the improvement strategy for the SuperPoint model.

References

[1] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In Proc. IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR'01), Hawaii, Dec 2001.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards,"ORB-slam: A versatile and accurate monocular slam system," IEEE Trans. Robot., vol. 31, no. 5,pp. 1147–1163, Oct. 2015.

[3] R. Mur-Artal and J. D. Tardos,"ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," IEEE Trans. Robot.,vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski,"ORB: An efficient alternative to sift or surf," in Proc. Int. Conf. Comput. Vis., 2011,pp. 2564–2571.

[5] D. Galvez-Lpez and J. D. Tardos,"Bags of binary words for fast place recognition in image sequences," IEEE Trans. Robot., vol. 28, no. 5,pp. 1188–1197, Oct. 2012.

[6] A. I. Comport, E. Malis, and P. Rives,"Accurate quadrifocal tracking for robust 3d visual odometry," in Proc. IEEE Int. Conf. Robot. Autom., 2007,pp. 40–45.

[7] C. Kerl, J. Sturm, and D. Cremers,"Dense visual slam for RGB-D cameras," in Proc. 2013 IEEE/RSJ Int. Conf. Intell. Robots Syst., 2013,pp. 2100–2106.

[8] R. Kmmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard,"G2o:A general framework for graph optimization," in Proc. IEEE Int. Conf.Robot. Autom., 2011, pp. 3607–3613.

[9] R. A. Newcombe et al.,"Kinectfusion: Real-time dense surface mapping and tracking," in Proc. 2011 10th IEEE Int. Symp. Mixed Augmented Reality, 2011, pp. 127–136.

[10] J. Engel, T. Schöps, and D. Cremers,"LSD-slam: Large-scale direct monocular slam," in Proc. Eur. Conf. Comput. Vis., 2014, pp. 834–849.