

論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Depth Estimation and Object Detection in Different Scales by Using PCA and Multi-pooling Layers
著者(和文)	郭宇翔
Author(English)	Guo Yuxiang
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12541号, 授与年月日:2023年9月22日, 学位の種別:課程博士, 審査員:熊澤 逸夫,山口 雅浩,鈴木 賢治,小尾 高史,篠崎 隆宏
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12541号, Conferred date:2023/9/22, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Depth Estimation and Object Detection in Different Scales by Using PCA and Multi-pooling Layers

Advisor: Prof. Kumazawa Itsuo
Submitted by: Guo Yuxiang
Date: August 14th, 2023
College: Engineering
Major: Information and Communications
Engineering



Tokyo Institute of Technology

Directory

List of acronyms	4
Abstract	5
Chapter 1. Introduction	7
1.1 Background survey on the depth inference	7
1.2 Background survey on the scale invariant recognition	8
1.3 Research objective	11
1.4 Summary.....	12
Chapter 2. Feature extraction and representation	13
2.1 Feature extraction	13
2.1.1 Statistics-based feature extraction.....	13
2.1.2 Model-based feature extraction.....	14
2.2 Feature representation	14
2.3 Feature evaluation.....	15
2.4 Features commonly used in image processing	17
2.4.1 Color features	17
2.4.2 Texture feature	17
2.4.3 Shape texture	18
2.4.4 Scale-invariant feature transform (SIFT).....	18
2.5 Feature map and feature vector	18
2.6 Feature invariance	20
2.7 Summary.....	20
Chapter 3. Theoretic base of the related research contents.....	22
3.1 Convolutional neural network	22
3.1.1 Overview of CNN	22
3.1.2 Basic principle of CNN.....	24
3.1.3 Loss functions	29
3.1.4 CNN basic framework	33
3.2 Principal component analysis (PCA).....	37
3.2.1 PCA introduction.....	37
3.2.2 The basic principle of PCA	38
3.2.3 PCA usage for image compression	40
3.2.4 PCA usage for feature determination	42
3.3 Discrete cosine transform (DCT)	44
3.4 Summary.....	46
Chapter 4. Object detection from monocular camera images with depth cues extracted by CNN	47
4.1 Introduction	47
4.2 Algorithm.....	50
4.2.1 Coarse-to-fine analysis	51
4.2.2 PCA for image processing	51
4.2.3 Edge Line Density Extraction by Discrete Cosine Transform	55

4.3 Depth cue derived by CNN	58
4.3.1 Convolutional Neural Network	58
4.3.2 Proposed Structure Framework	59
4.3.3 CNN Diagram and Parameters	66
4.4 Evaluation of the proposed method	68
4.4.1 Experimental setup and testing	68
4.4.2 Testing for the proposed method	69
4.4.3 Practical application study for vehicle blind spot detection	70
4.5 Summary and research in next step	72
4.5.1 Summary	72
4.5.2 The research in next step	73
Chapter 5. Object detection in different scales by using PCA and multi-pooling layers	75
5.1 Introduction of object detection	75
5.2 Related research	75
5.3 Proposed algorithm structure	78
5.3.1 Brief of the CNN outline	78
5.3.2 Description of multi-pooling layers	79
5.3.3 Feature of scale invariance derived from PCA process	81
5.4 Network model verification for scale variant object recognition	86
5.5.1 Algorithm structure	86
5.5.2 Model training and testing	89
5.5.3 Network performance comparisons	97
5-5. Discussions	102
5-6. Summary	103
Chapter 6. Conclusion remarks	104
References	105
Research achievements	110
1. Journal Papers	110
2. Invited Conference presentation	110
3. Patents	110
Acknowledgements	111

List of acronyms

ADAS: Advanced driver assistant system
ASPP: Atrous Spatial Pyramid Pooling
BSD: Blind spot detection
CNN: Convolutional Neural Network
FCN: Fully Convolutional Networks
MAE: Mean Absolute Error
MSE: Mean Squared Error
PCA: Principal Component Analysis
R-CNN: Region-based CNN
RFP: Recursive Feature Pyramid
RPN: Region Proposal Network
SfM: Structure-from-Motion
SGD: Stochastic Gradient Descent
SIFT: Scale-invariant feature transform
SPPNet: Spatial Pyramid Pooling Networks
SSD: Single Shot Detector
SVM: Multiclass Support Vector Machine Loss
MRF: Markov random field

Abstract

Depth information can be inferred from a monocular camera 2-D image. Depth information is given as the classification of “near” or “far” region when two blocks in the image are compared with respect to their distances. As the focus principle of perspective view, the texture cue is rich around the focus spot, and the blur cue is surrounding the monocular camera spot. Therefore, the depth cue is estimated by the comparison of texture cue and blur cue.

PCA (principal component analysis) process is conducted to extract the edge lines corresponding to eigenvector orientation. The proposed classified edge line orientation is very helpful in extracting local depth cues because of the enhanced effectiveness of the spatial frequency analysis process. The depth information of the entire image can be obtained from the relative depth comparisons of all blocks. To adapt to the real world application, this depth estimation method is verified by BSD (blind spot detection) function implementation for ADAS (advanced driver assistant system) with acceptable detection performance under the limited conditions of cost-effective monocular image processing technology. In the absence of disparity information, the relative depth information obtained by this method is more reliable.

The object scale on 2-D image is changing along with the depth. The larger the depth is, the smaller the object scale; and vice versa. Therefore, it is necessary to take into account the change of object scale when CNN recognizes the scale variant object. Many of the object recognition methods at different scales are benefited from the features extracted from deep CNN structure. One of the most widely used structures is the pyramid structure, in which the image is resized to be processed in multiple levels. Different object scale belongs to a level in the pyramid. And no matter which pyramid level an object belongs to, it is subject to the object class. The coarse-to-fine pyramid can also support feature extraction of texture cue and blur cues.

The eigenvalue obtained through PCA can represent the object point distribution, i.e. the object shape. The eigenvalue changes along with the scale variation. In order to get the scale invariant shape feature, the eigenvalue normalization is applied to remove the influence of the object scale dependency. Through theoretical calculation and experiment verification, it has been proved that the normalized eigenvalues can represent the scale invariant shape features of the object. Consequently, in order to enhance the performance and robustness of scale variant object detection, a novel

network structure is proposed in which the module of multi-pooling structure and PCA processing is embedded before fully connected layers. This process is equal to apply noise filter to the input image data. The feature map becomes more focused on the activated object shape. In this case, the feature of scale invariance is embedded through module of the parallel PCA processing with multi-pooling layers, and then fully connected layers produce the Softmax accuracy prediction for scale variant object recognition. This modified deep CNN model is trained with the dataset of scale variant vehicle images captured from road vehicle traffic scene. In the experiments, different network structures are implemented: VGG-16, VGG-16 with SPP, VGG-16 with multi-pooling module and VGG-16 with proposed structure. The detection performance of variant vehicle scale is compared between basic VGG16 network and the spatial pyramid pooling network as well as the proposed network structure of module with PCA and multi-pooling layers. The other networks of the current popularity are also tested by our dataset. These networks are ViT-L/16, Nasnet-A(6) and Resnet-152. Through these comparison studies, it is proved that the proposed module achieves the state-of-the-art result. Moreover, we embedded the proposed module into YOLO network, and the detection performance is evaluated by mean average precision. The experiment is carried out by using dataset PASCAL VOC. The YOLO network embedded by our proposed module achieved better accuracy than YOLO itself. Through the comparisons, the proposed module promotes the feature extraction of scale invariance and enhances the recognition accuracy for scale variant object classification. The testing results demonstrated that the proposed network structure can achieve the comparable detection accuracy as per the state-of-the-art result. The module of multi-pooling loop parallel to PCA processing loop can be embedded to other deep network structure and the comprehensive capability can be expected.

Chapter 1. Introduction

1.1 Background survey on the depth inference

Human vision has the capability of inferring object image color, size and distance simply, as the human being growing up with many years' trainings on environment perception. Similarly, computer vision needs also much more efforts to complete the same tasks based on the captured image data. Due to 3-D data acquisition device is very expensive, it is necessary to reconstruct the available 2-D images into 3-D contents and obtain the depth information. In general, the 3-D image inferred from a single 2-D image is a topic of how to generate depth map information from the image. Because 2-D image generation is a process of projection from 3-D to 2-D, the depth information is lost at the time of image capture by monocular camera. However, this depth information can be inferred back from the features presented in the image. Therefore, the extraction methods of the depth cues are still the interested research topics.

A lot of studies on the depth cue perceptions from a monocular view have been conducted by many researchers. Ko, et al.^[1] proposed a depth estimation method through image segmentation with assumption that attention image cues discriminate foreground and backgrounds. It can estimate a relative depth map and then refine the depth map by post-processing. A stereoscopic image is generated by calculating the parallax values of each region using the generated depth-map and the input image. Ashutosh, et al.^[2] presented a discriminatively trained Markov random field model for depth estimation from single monocular image. The model uses monocular cues at multiple spatial scales. It is demonstrated that the algorithm gives good 3-D depth estimation performance on a variety of images. The survey [3] investigated the existed 2-D to 3-D conversion algorithms developed in the past 30 years by various computer vision research communities. It described and analyzed the algorithms that use a single depth cue and several promising approaches using multiple cues, establishing an overview, and evaluating its relative position in the field of conversion algorithms. Liu et al.^[4] proposed a depth estimation method based on semantic segmentation and semantic labels. It could provide more perceptual way to assign the depth cues; however, the proposed technique needs specific learning stages. Huang et al.^[5], presented a depth extraction method based on motion and geometric information for 2-D to 3-D conversion. The motion-based depth map and the geometry-based depth map are integrated into one depth map by a depth fusion algorithm. The geometry-based depth

map is generated by Hough transform with edge detection information. But the traditional Hough method is difficult to be applied to complex geometric environments. In reference [6] the depth map information is generated to extract the line features from a single image. This is used as an input source of extracting geometric cue, i.e., vanishing point. The moving object scenes are composed of parallel lines. These lines in real scenes appear to converge with distance in perspective image. The linear perspective is an important depth cue to generate the final depth map.

With regarding to the depth information estimation by the feature of image blur and texture, the previous research is conducted in using image blur as a cue for depth estimation. In reference [7], the authors proposed the technique of estimating depth-map in a single image. This technique uses both vanishing point and super pixels as geometric and texture cues for 2-D to 3-D image conversion processing. The texture cues are used but need to extract vanishing lines. In reference [8], the authors proposed the depth estimation from focus as the first deep learning solution. This method utilized the blur cues as the feature for CNN(convolutional neural network) learning. It reduces the depth error regarding to the existed researches which are based on blur cues. Mather^[9] described that the images of 3-D scenes inevitably contain regions that are spatially blurred due to depth-of-focus limitations in the image taken by monocular camera. And this blur variation acts as an effective cue to depth. If one image region contains sharply focused texture, and the other contains blurred texture, then the two regions can be perceived at different depths according to the spatial frequency discrimination. The comparison study showed that blur information should be more effective for larger distances.

As the summary, according to the very popular monocular camera applications in real life with the benefit of cost-performance, the progress on the distance information derivation is still the research focus. The features of image blur and texture are important to infer object distance information. The efforts of making an optimal deep CNN structure are necessary.

1.2 Background survey on the scale invariant recognition

Object detection is the task of detecting object instances in an image, and its goal is to determine whether there are the instances of objects from given categories, like cars, bicycles, pedestrians, cats and so forth. Object detection supports a wide range of

applications, and it can be classified as the detection of specific instances and the detection of broad categories. The first type aims to detect instances of a particular object, essentially subject to a matching problem. The second type is to detect instances of some predefined object categories. So far, much of the effort in the field of object detection has been focused on the detection of a single category, typically human faces, and pedestrians^[10], or a few specific categories to meet the real application needs.

In recent years, deep CNN has played an important role in many computer vision tasks^[11]. Deep CNN for object detection covers many aspects of generic object detection: detection frameworks, object feature representation, object proposal generation, context modeling, training strategies, and evaluation metrics. As the state-of-the-art, the methods in computer vision have increasingly utilized deep CNN architectures, with some of the most successful models employing up to thousands of layers^[12].

Another popular technique in the last two years is multi-resolution detection^[11], i.e., by detecting objects of different scales at different layers of the network. Since a deep CNN naturally forms a feature pyramid during its forward propagation, it is easier to detect larger objects in deeper layers and smaller ones in shallower layers. Multi-reference and multi-resolution detection have now become two basic building blocks in the state-of-the-art object detection systems.

The detection accuracy depends heavily on its feature extraction from the network layers. The deep CNN architecture used in object detection is devoted to improving object feature representations, such as developing invariant features to accommodate geometric variations in object scale, pose, viewpoint and performing multi-scale analysis to improve object detection over a wide range of scales. Deep CNNs can learn powerful feature representations with multiple levels of abstraction directly from raw images. The good feature representations are of primary importance in object detection. Since a deep CNN computes its feature hierarchy layer by layer, the sub-sampling layers in the feature hierarchy lead to an inherent multi-scale pyramid, producing feature maps at different spatial resolutions. The image pyramid is simple and effective, helping to enlarge small objects and to shrink large ones^[13,14]. It is computationally expensive but is nevertheless commonly used during inference for better accuracy. Scale-invariance can be achieved using an image pyramid, which contains normalized input representations of object instances in one of the scales in the image-pyramid. The

scale-invariance can be naively expressed as the object looks the same at every scale, or no matter zoom-in or zoom-out the object view, its features remain unchanged^[15]. Scale variation across object instances is one of the major challenges for the object detection task.

Small object detection in principle makes the detection task very challenging. The improvement is to enhance the detection accuracy by exploiting multiple CNN layers, i.e., multi-scale object detection at multiple layers or with combined features of multiple layers. The combined feature is more descriptive, and is more beneficial for localization and classification, but at increased computational complexity^[16]. Object detection requires robustness to real-world variations, and the variant scale is dominant scene especially in viewing moving objects. As per the real-world application, e.g., in autonomous driving, due to the perspective projection effect, vehicles far away are very small, while the near ones are relatively larger in the captured image. As a result, performing a uniform feature alignment across all scales for domain adaptation would be difficult and this requires the object detection with the attribute of scale-invariance.

In dealing with the problem of detecting objects at different scales, many approaches are tried to achieve the accurate results with less computation cost. In the literatures^[13,14,19,20,21], pyramid structure is intensively used to extract features and it is applied to support object detection at different scales within the image. Image pyramid is usually obtained by down sampling or up sampling the original image multiple times. And then we can detect objects at different resolutions and get more comprehensive object detection results. Due to the dynamic real world scene is keeping changing, especially the moving objects, the captured images may be collected at different scales. Image pyramid has a wide range of applications in multi-scale object recognition and is useful to analyze and process images at different scales. Image pyramid is usually applied at the entrance of deep CNN^[13,14]. An image pyramid is a multi-scale representation of an image, where the original image is progressively down sampled to create a series of images at different resolutions. The pyramid typically starts with the original image at the highest resolution and progresses downwards to coarser resolutions. Eigen^[22] and Pinheiro^[23] apply multi-scale inputs sequentially from coarse-to-fine, while the reference studies [24, 25, 26] directly resize the input for several scales and fuse the features from all the scales.

The coarse-to-fine approach is a common strategy for using image pyramids in computer vision applications. The coarse levels of the pyramid provide a low-resolution overview of the image, which can be used to quickly identify large-scale features or regions of interest. The finer levels of the pyramid provide progressively more detailed information, which can be used to refine the analysis of the image and localize features with higher precision.

The reference study [19] proposed a recursive feature pyramid network (RFP), which use single scale features and is computed like featurized image pyramid. Feature pyramid is a basic component for detecting objects at different scales. By exploiting inherent multi-scale, pyramidal hierarchy of deep CNN can be used to construct feature pyramids with marginal extra cost. A top-down architecture with lateral connections is developed for building high-level semantic feature maps at all scales.

CNNs are typically designed to take in fixed-size inputs, which limit their ability to recognize objects at different scales^[27,28]. Due to the fixed input image size requirement, image data pre-conditioning becomes necessary.

Principal component analysis (PCA)^[17,13,14] is a widely used technique in image processing and computer vision for feature extraction and dimensionality reduction. It can be applied to detect scale variant objects in images and is used to extract a set of features from the preprocessed images. These features are computed from the covariance matrix of the image pixels and represent the directions of maximum variance in the image. Once the feature vectors have been reduced in dimensionality, they can be used to detect objects in the image. To detect scale variant objects, PCA can be applied at multiple scales, or image resolutions. This allows the algorithm to capture features at different levels of detail, and to detect objects of different sizes or scales.

In the real world, the object image presents different scales due to observing direction and distance. It is meaningful to recognize the same object at different scales.

1.3 Research objective

Depth information is a very important cue for computer to learn the 3-D information from 2-D image. In order to extract the depth cue information, the parallel structure is formulated by the texture cue path and blur cue path. The texture cue can be derived from the process of principal component analysis and discrete cosine

transform. The blur cue is derived by discrete cosine transform only. Through these steps, CNN is applied to complete the feature learning and output the depth predictions.

Depth variation causes the object scale variation in the image. The accuracy of object recognition is related to the scale. The features of scale invariance are required for object recognition tasks through CNN. The normalized eigenvalue of object represents the scale invariant shape features and is used as the scale invariant feature. Therefore, the module of PCA with normalized eigenvalue parallel to multi-pooling layers is proposed for the purpose of feature extraction of scale invariance. And this module is embedded to CNN to form a new network main stream structure.

The performance of depth information prediction and the scale invariant feature learning are evaluated by using real world traffic image data. Based on the study artifacts, some conclusions are summarized.

1.4 Summary

This chapter describes the research background, motivation and challengeable and also meaningful topics with regarding to the object detection method for depth inference from the popular monocular camera image, and detection method for the scale variant object in 2-D image. Monocular camera applications with the benefit of cost-performance are dominant in the real world. The features of image blur cue and texture cue are important to infer object depth or distance information. Depth variation causes the object scale variation in the image. The features of scale invariance are required for object recognition tasks through CNN. The typical situation of the real-world traffic scenario is the scale variant moving vehicle. To early detect such scale variant object make much sense for safety reasons. So the research objective is focused on the depth estimation and object detection in different scales.

Chapter 2. Feature extraction and representation

A feature is traditionally defined as a logical unit of behavior specified by a set of functional requirements^[29,30], although the definition of feature itself is the subject of study^[31]. A machine learning or deep CNN learning problem starts with features building from images. The quality of features directly affects the final model results.

Feature extraction is an important element of image processing. Convolutional neural networks are powerful models that achieve impressive results for image classification. The pre-trained CNN models are useful for generic feature extraction^[32,33]. Especially the proposed various pyramid feature fusion methods further improve the performance of CNN feature extraction, but at the same time they lead to more complex CNN structure and higher requirements for the sample quantity and computational performance, which has an impact on the application of object detection using high-resolution images. PCA and DCT(Discrete Cosine Transform) are effective methods for feature extraction, but both also add additional computational overhead, and it is worth exploring how to effectively combine PCA, DCT and CNN to improve the effectiveness and efficiency of feature extraction.

The following sessions provide the basic concept definitions and comprehensive points of view in the interested study field. Conceptually, the feature construction usually includes feature extraction, feature representation and feature evaluation. The brief overview is given in the below sections.

2.1 Feature extraction

Feature extraction refers to selecting features that are useful for the final machine learning or deep CNN learning task from all the features. Depending on whether the model is involved in the process of feature extraction, the method of feature extraction can be divided into statistics-based extraction and model-based extraction. By default, all feature values are dimensionless, or have been normalized in the discussion.

2.1.1 Statistics-based feature extraction

If each feature is regarded as a random variable, the random variable may take different values at different sample points. Statistical methods can be used to extract features based on the statistical results of the sample set. There are two main criteria for extraction, one is the degree of dispersion of the value of the feature itself; the other is the degree of correlation between the feature and the result to be predicted. The common

statistics methods are variance, correlation coefficient, hypothesis testing, and mutual information.

2.1.2 Model-based feature extraction

Model-based feature extraction can be selected directly based on model parameters. For models with a linear structure, such as linear regression model and logarithmic linear models (such as logistic regression, maximum entropy, linear chain conditional random fields, etc.), the importance of corresponding features can be directly measured according to the size of the weight parameters.

Because of the linear structure of the model, if the absolute value of the parameter corresponding to a feature in a certain dimension is large, the feature of this dimension is relatively important; contrarily, if the absolute value of the parameter is small, the feature is relatively unimportant.

2.2 Feature representation

In this session, the feature representation (or feature encoding) is discussed with regarding to how to describe specific objects from these confined dimensions. Feature representation needs to be considered from a complete machine learning or deep CNN learning task. After the feature is extracted, the task of feature representation is to abstract each sample into a numerical vector by deep CNN learning model. Therefore, feature representation must take into account both feature attributes and model requirements.

Features can be simply divided into continuous type and discrete type according to their value types. Required features include type matching, feature accuracy, feature completeness, and convenient model training. For models based on mathematical operations, the features must be converted to numeric types, and there must be corresponding processing logic for missing values. The model accuracy is roughly divided into "high", "medium" and "low", although not precise enough. Completeness refers to whether a feature can describe an object in as many ways as possible. This has a lot to do with the used model. Simple models want the features to be complex enough to take into account the required higher-order features. But for complex models, such as neural networks, the model structure itself has a certain feature cross function, so high-order features can be ignored. Convenient model training is from the perspective of solving model parameters. If the features are normalized, it can effectively avoid the

formation of "canyons" in the solution space, thereby speeding up the parameter solving process. The technical problem of feature representation is to process the features continuously or discretized according to the needs of the model. Continuous features do not need to be serialized, and the value of the feature can be used directly, and at most, a normalization processing is sufficient. Due to various factors in the data source and processing process, there might be problems with missing values for feature extraction formulations. For most models, missing values need to be dealt with through screening and filling.

2.3 Feature evaluation

Feature evaluation refers to the overall evaluation of the extracted features. In the process of feature extraction, each dimension of the feature needs to be evaluated to extract relatively more important features. Regarding to the extracted feature dimension, the feature is encoded according to the original data, and then the feature is obtained.

The final target of feature extraction and representation is to provide predictions to the model. Before analysis, it is necessary to give priority to the features that perform well on the model. Therefore, before the final quantitative analysis, it is necessary to evaluate the feature criteria in terms of feature coverage, feature dimensions, and qualitative analysis.

Coverage refers to the proportion of images that can successfully generate features to the entire image data set. Different methods of feature extraction and generation have different coverage rates. The features of the image are proposed based on the "label" attached to the content. Image tags are the embodiment of content and a typical way to generate features based on content. The coverage of the method depends on the coverage of the labels.

In actual engineering implementation, the dimension of features is very important because the dimension of the feature is related to the expressive ability of the feature, the applicable model, and the calculation complexity. If the feature dimension is very low, it is obvious that the expressive ability of the feature is limited; if the feature dimension is very high, the calculation efforts will be upgraded (the calculation cost is related to the complexity of the model), and it is easy to cause the problem of dimension disaster. Therefore, the dimensionality of features should be compatible with the computing framework and data scale. The first is that the feature dimension must match the capabilities of the computing framework. Then the feature dimension should match

the data size.

Qualitative analysis of feature interpretability can be performed based on understanding of the real world and contextual perception information. Features can be fed into model training and then quantitatively analyzed based on model results.

Features are fed into the model, and the results are quantified, which is very close to the concept of model evaluation. For a trained model, it is important to evaluate its generalization ability. For classification problems, you can examine the accuracy rate, recall rate, and other indicators on an independent test set; for regression problems, there are also conventional evaluation indicators such as MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error).

In order to find the problem of poor feature performance and find the direction of improvement, first analyze how to get the final quantitative analysis index (such as RMSE) for the purpose of issue reverse troubleshooting. First, features need to be extracted from the data source. In this process, the quality of the data source and the feature extraction scheme will affect the final features. Model selection and model training will affect the final analysis indicators. Therefore, when the feature performance is not good enough, it is necessary to reverse check the problem according to this process.

There is no such a model that can cope with every situation in deep CNN learning. The appropriate model should be selected according to the characteristics and data volume. For example, KNN (K-Nearest Neighbour: Machine Learning algorithms based on supervised learning technique) performs very well on low-dimensional problems, but is not suitable for high-dimensional features due to the problem of dimensional disaster.

When the data source is determined, the factors affecting feature quality are mainly feature extraction and feature encoding. In terms of feature extraction, it is necessary to consider whether the extracted features are complete and how redundant they are.

One more concerning is data source issues. Feature construction based on image labels, relying on labeled data. If the labels are inaccurate, or the data is contaminated, the quality of the features is difficult to be guaranteed. The big effort for labeling is the foundation of high-quality results.

2.4 Features commonly used in image processing

The commonly used image features in image processing can be summarized as: color features; texture features; shape features; spatial relationship features.

2.4.1 Color features

The color feature is a global feature that describes the surface properties of the scene corresponding to the image and the image area. The general color feature is a feature based on pixels. At this time, all pixels belonging to the image or image area have their own contributions. Since the color is not sensitive to changes such as the direction and size of the image or image area, the color feature cannot capture the local characteristics of the object in the image well. Color histogram is the most common used method to express color features, and its advantage is that it is not affected by image rotation and translation changes. With the help of normalization, it is also not affected by changes in image size. Its disadvantage is that it does not express the information of the color space distribution.

2.4.2 Texture feature

Texture feature is also a global feature, which describes the surface properties of the scene corresponding to the image or image region. However, since texture is only a characteristic of the surface of an object and cannot fully reflect the essential properties of the object, high-order image content cannot be obtained only by using texture features. Different from color features, texture features are not pixel-based features, which require statistical calculations in regions containing multiple pixels. In model matching, such regional features have greater advantages, and will not fail to match successfully due to local deviations. As a statistical feature, texture features often have rotation invariance and have strong anti-noise ability. The obvious disadvantage of texture features is that when the resolution of the image changes, the calculated texture may have a large deviation. In addition, because it may be affected by light reflection, the texture reflected from the 2-D image is not necessarily the real texture of the surface of the 3-D object. For example, reflections in water, smooth metal surfaces, the effects of mutual reflections, etc. will all cause changes in texture. Since these are not properties of the object itself, these spurious textures can be misleading for feature extraction. When extracting texture images with large differences in thickness and density, using texture features is an effective method

2.4.3 Shape texture

Various extraction methods based on shape features can effectively use the target of interest in the image to extract, but they also have some common problems, including (1) the current shape-based extraction method still lacks a relatively complete mathematical model; (2) if the target is deformed, the extraction results are often unreliable; (3) many shape features only describe the target, and the local properties must fully describe the target, which will have high requirements for computing time and storage capacity; (4) Targets reflected by many shape features.

The redundancy of shape information is not completely consistent with human intuitive feeling, or the similarity of feature space is different from the similarity perceived by human visual system. In addition, the 3-D object represented by the 2-D image is just the projection of the object on a certain plane in space. The shape reflected from the 2-D image is often not the real shape of the 3-D object. Due to the change of the viewpoint, various distortions may occur.

2.4.4 Scale-invariant feature transform (SIFT)

SIFT^[34] features are scale and rotation invariant. It detects and matches features across different scales and orientations. This makes the SIFT algorithm highly effective in handling images with variations in object size, viewpoint, and orientation. SIFT descriptors capture unique and distinctive characteristics of image regions, enabling robust feature matching even in the presence of partial occlusion and cluttered backgrounds. The distinctive nature of SIFT features makes them highly suitable for tasks such as object recognition and image retrieval.

But SIFT is sensitive to image noise, blurring, and lighting which can degrade the accuracy of key-point detection and descriptor matching. In the presence of significant noise or blur, the performance of SIFT is not good enough. SIFT may not capture high-level semantic information or context. This can limit its performance in tasks that require a deeper understanding of the image content.

2.5 Feature map and feature vector

In image processing tasks, the feature maps in CNNs represent visual patterns and features such as edges, corners, shapes, and textures in the input image. These features are learned through convolutional and pooling layers, and are progressively combined

to form higher-level features that correspond to object parts and eventually the entire object in the image.

The feature of an image is the data expression form obtained by processing the digital image for a specific purpose. It includes two levels, one is the structure level, and the other is the semantic level. The structural level refers to the data representation form of features, such as eigenvalues (scalars), eigenvectors, and eigen-maps (matrices). The semantic level refers to the definition of visual attributes of features, such as color features, corner features, edge features, texture features, shape features, and so on.

The digital image of a specific color space is represented by a multidimensional matrix, noted as $I \in \mathbb{R}^{W \times H \times C}$, W and H are the number of columns and rows of the matrix corresponding to the number of width and height pixels of the image, respectively, and C denotes the number of channels. Provided that the transform operator is $f \in \mathbb{R}^{k \times k \times C}$, and its number is c . Each transform operator f slides continuously on I at defined step, and at each position with the corresponding element of I . Through these processing, finally get the matrix $F \in \mathbb{R}^{w \times h \times c}$, which is called as F the **feature map** of the image I , c is the number of channels of the feature map. Like the concept of image resolution, the feature map consists of wh pixels, where wh is called the resolution of the feature map. Each pixel corresponds to a c -dimensional vectors, all c -dimensional vectors of the feature map are connected to form a $w \times h \times c$ -dimensional vector, which is called the **feature vector** of I . The relationship between image, feature map and feature vector are shown in Fig. 2-1. The operation of feature map follows the operation rules of multidimensional matrix, such as summation, dot product, etc. Another common feature map operation is to stack feature maps of the same resolution with different

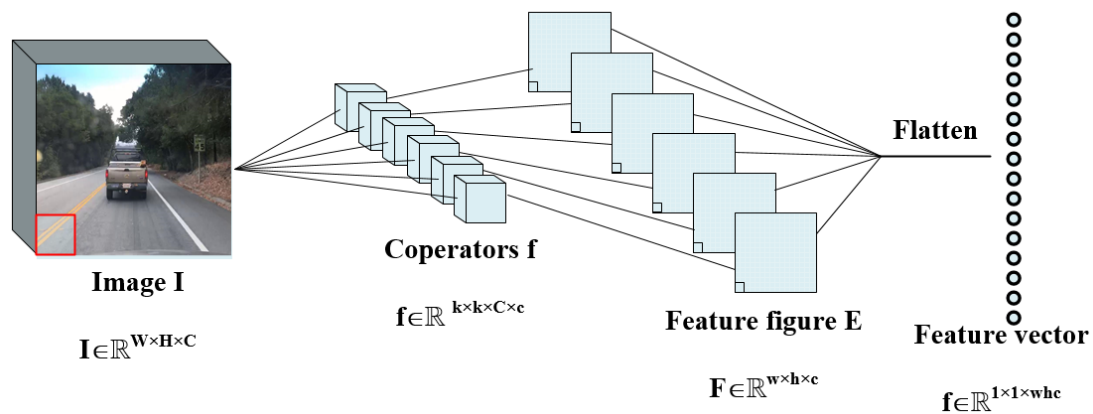


Figure 2-1 Relationship between image, feature vector and feature map

number of channels together on a channel. The process of putting the feature maps through certain operations to get a new feature map is called feature fusion.

2.6 Feature invariance

The property that the eigen-characteristics of image retain unchanged is called as invariance. The same type of objects could present many differences; the different types of objects would have many similarities, and the specific objects of the same type can also present different shape due to motion (perspective changes, scale changes, pose changes, background, and illumination changes, etc.).

The common feature invariances can be thought as rotation invariance, scale invariance, and affine invariance. Rotation invariance is to define the direction of the feature, and then describe the feature in the same direction to achieve rotation invariance through rotation normalization. To achieve scale invariance, it is necessary to add a scale factor to the features. The scale invariance can be achieved by unifying the scale. This process is called scale normalization.

Local invariant features refer to the description of local features that remain invariant to various changes in the image, such as geometric transformation, photometric transformation, convolution transformation, and viewing angle changes. The basic idea of local invariant features is to extract the essential attribute features of the image content, and these features have nothing to do with the specific representation of the image content. Scale-invariant feature transform (SIFT)^[34] is an algorithm used to detect and describe local features in images. It searches for extreme points in the spatial scale and extracts their position, scale, and rotation invariance.

Local invariant features can not only obtain reliable matching in the case of large changes in observation conditions, occlusion, and cluttered interference, but also effectively describe image content for image retrieval and object recognition. Local invariant features can overcome the need for semantic-level image segmentation.

2.7 Summary

This chapter provides the basic technical comprehension on feature extraction and representation. The contents include the description of the methods and ideas of feature extraction from the perspectives of statistics-based and model-based approaches. The overview gives us the fundamental knowledge information. The local feature is used by the selected feature map with sliding window processing for object depth estimation.

The feature concept of texture and blur cues is explored in the context. The normalized invariant feature is derived by using principal component analysis. The feature extraction method of scale invariance for the same object at different scales is developed.

Chapter 3. Theoretic base of the related research contents

3.1 Convolutional neural network

3.1.1 Overview of CNN

Research on convolutional neural networks can be traced back to the Neocognitron model ^[35,36,37] proposed by the Japanese scholar Kunihiko Fukushima, who designed a neural network named “Neocognitron” after the visual cortex of living creatures. Neocognitron’s S-layer-C-layer combination is capable of feature extraction and filtering, which partially implements the convolution layer in convolutional neural networks. In 1989, Lecun proposed a neural network in which the structural form of convolutional operations and the rules of operations were clearly defined, which is the pioneer of modern convolutional neural networks. Later around 1998, Lecun et al^[38] proposed the widely circulated LeNet-5, which was successfully applied to the handwritten digit recognition business in postal and banking and had a profound impact on the research of neural networks later.

The development of truly deep convolutional networks started in 2012 with the AlexNet designed by Alex^[39]. The network is divided into 8 layers, including 5 convolutional layers and 3 fully connected layers, with about 61M parameters (249MB of memory) and performs 1.5B high precision operations to classify one image. These numbers are even higher for deeper CNNs e.g., VGG. It proposed a new activation function ReLU, Dropout mechanism and group convolution. After that, deep convolutional neural networks were proposed one after another, the most representative ones are VGG-16 proposed by Simonyan and Zisserman in 2014^[40], GoogLeNet proposed by Christian Szegedy in 2014 with 48 layers^[41]. As the number of layers of CNN increases, i.e., the depth increases, the gradient of the network will disappear or diverge during the training process and it is difficult to converge. By normalizing the initialization, the input of each layer is normalized, so that the depth of the network that can converge is improved. However, as the depth of CNN increases again, the network accuracy does not improve effectively (the error curve becomes flat) but decreases (the training and testing errors become larger compared to the network with fewer layers) (also called network degradation). In 2015, He et al. proposed the residual network ResNet^[42], which uses jump connections (shortcut) to effectively solve the degradation problem, and the network depth reaches 101 layers. In 2017, Hu et al. proposed the channel attention module and used it to design SENet^[43] which won the last ImageNet

2017 competition for the image classification task. The simple structure of the channel attention module is easy to embed into existing networks and keeps the growth of additional parameters and computations minimal, and has been widely used in subsequent semantic segmentation networks. In 2017, several lightweight network structures being proposed, including Google's MobileNet^[44] series. The first separable convolution structure (depth wise separable convolution) was proposed by Google in its MobileNet series, and Iandola et al.^[45] in their SqueezeNet, a compression module was proposed. Based on the above generic classification networks, many innovative network structures have emerged for semantic segmentation and object detection. Fig. 3-1 summarizes the classical approaches in classification (middle column), object detection (right column), and semantic segmentation (left column) for more than 20 years from 1998 to 2020. Deep learning methods are shown in solid boxes in Fig. 3-1, and traditional machine learning methods are shown in dashed line boxes. The bold font indicates important innovative ideas.

As we can see from Fig. 3-1, since 2012, deep learning methods have dominated in almost all aspects, with RCNN and YOLO family being the most popular for object detection, and FCN and SegNet being the most influential for semantic segmentation. Among the deep learning algorithms for object detection, R-CNN is the one that successfully uses deep learning for object detection^[46]. It is a two-stage “region candidate plus object detection” algorithm framework, where the letter R stands for region proposals. The R-CNN eliminates the sliding window operation in the traditional CNN algorithm framework to improve the efficiency of the algorithm, and gradually evolves into the RPN (Region Proposal Network) in the later series, and proposes the anchor box for border regression method. YOLO series is the so-called single-stage object detection algorithm^[47], and its network framework is like RPN. In processing the input image, it is rasterized to form the cell, and the size and position of the cell are used to directly train the network to regression the border of the target object. There is no region candidate process, so it is called single-stage object detection method. YOLO simplifies the complexity of network implementation, improves the computational efficiency of the algorithm, and can realize end-to-end training. In the later improved version, Anchor is also introduced into the algorithm framework to further improve the detection accuracy of the algorithm.

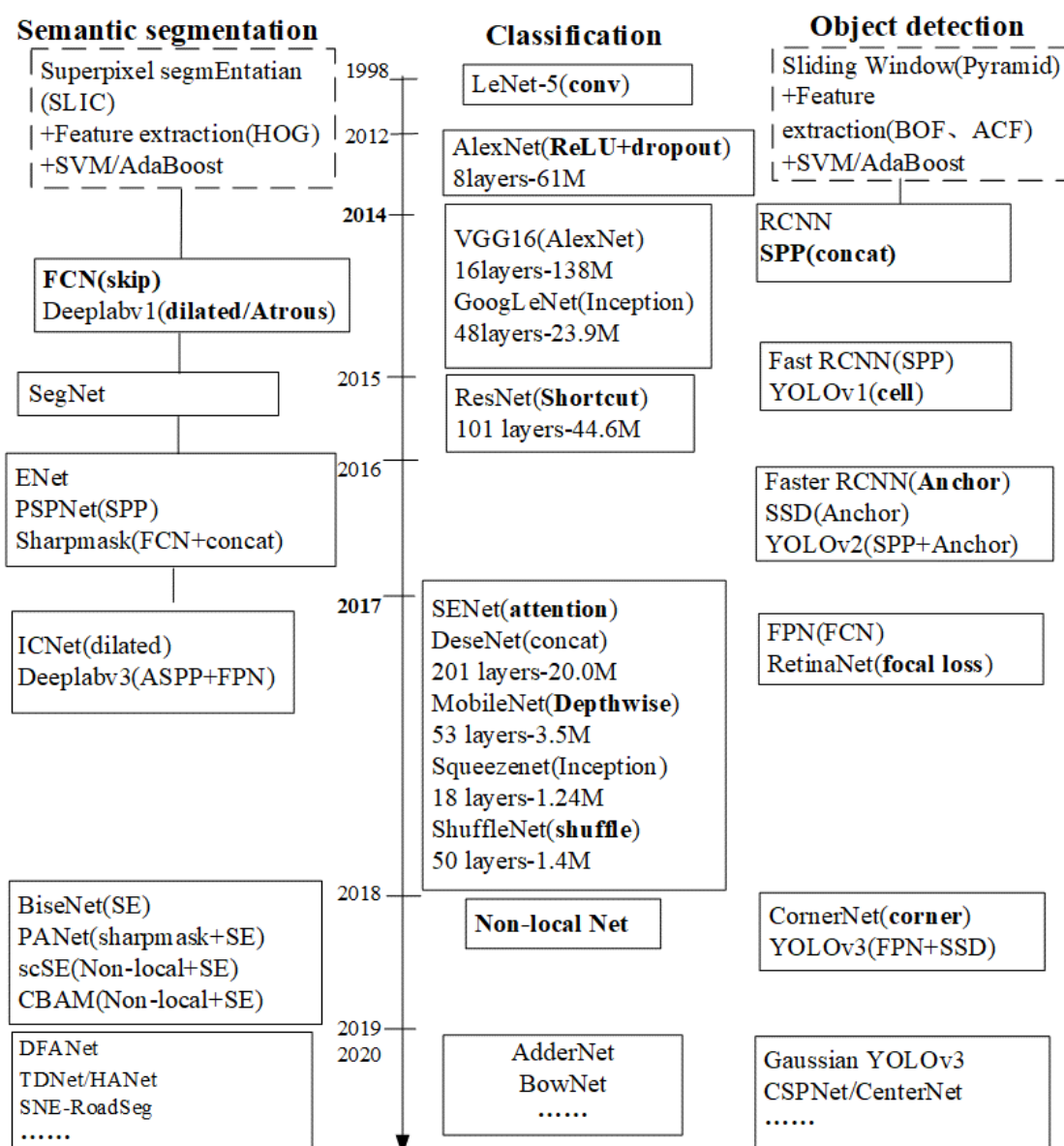


Figure 3-1 Development process of CNN

3.1.2 Basic principle of CNN

Convolutional neural networks consist of three main types of layers: **convolutional layers**, **pooling layers**, and **fully connected layers**, as shown in Fig. 3-2, where the convolutional and fully connected layers perform transformations on the input by using activation functions. Convolutional and pooling layers in a convolutional neural network perform feature extraction on the input data, while fully connected layers perform non-linear combination of the extracted features to produce the output. The convolution kernel in the convolutional layer contains weight coefficients, while the pooling layer does not contain weight coefficients. Therefore, the pooling layer is usually not considered as a separate layer in the literature.

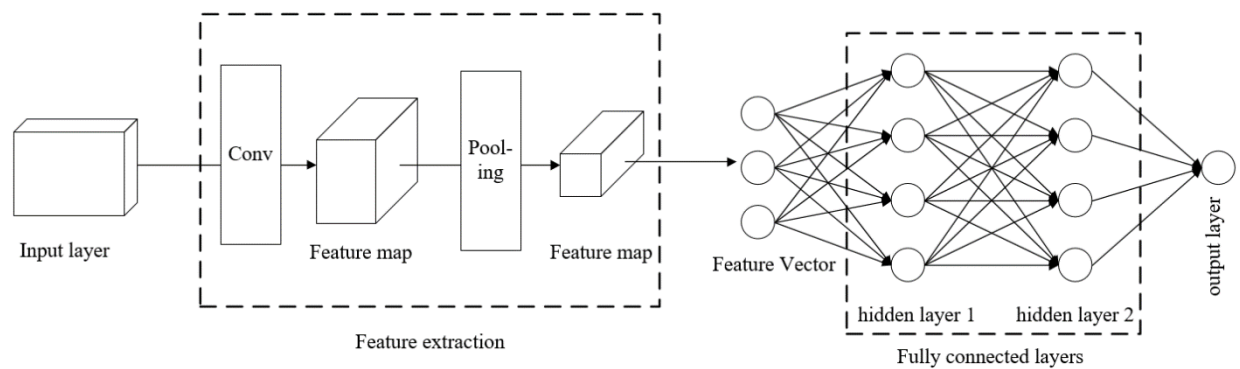


Figure 3-2 Basic hierarchical structure of CNN

The parameters of the convolutional layer include the size of the convolution kernel, the step size, and the padding, which jointly determine the size of the output feature map of the convolutional layer and are the hyper parameters of the convolutional neural network^[48]. The size of the convolution kernel can be specified as any value less than the size of the input image. The larger the convolution kernel is, the more complex the input features can be extracted. The step size defines the distance between two adjacent sweeps of the kernel. If the step size is 1, the kernel will sweep through the feature image elements one by one. If the step size is n , the kernel will skip $n-1$ pixels in the next scan. As the convolutional layers are stacked, the size of the feature map gradually decreases. Padding is a method to artificially increase the size of the feature map before it is passed through the convolution kernel to offset the effect of size shrinkage in the calculation. The common padding methods are zero-padding and replication padding.

The inclusion of the activation function in the convolutional layer makes the network model nonlinear and improves the capability of the network to extract complex features. Sigmoid function was common in early neural networks, but it has been gradually replaced by the simpler ReLU function. Other ReLU-like variants include leaky ReLU (LReLU) with slope, parametric ReLU (PReLU), etc.^[49].

After feature extraction in the convolution layer, the output feature map is passed to the pooling layer for feature selection and information filtering. The pooling layer contains a predefined pooling function that replaces the results of a single point in the feature map with the feature map statistics of its neighboring regions. Mean pooling and extreme pooling are often used pooling methods in the design of convolutional neural networks, both of which preserve the background and texture information of the image at the expense of losing some information or size of the feature map^[50]. In addition, FFT-based pooling methods can be used along with convolution to build FFT-based convolutional neural networks^[51].

The fully connected layer is located at the last part of the implicit layer of the convolutional neural network. It is equivalent to the implicit layer in a traditional feed-forward neural network. The feature map loses its spatial topology in the fully connected layer and is expanded (flattened) into vectors and passed through the excitation function. Adding specific solution oriented features before the fully connected layers can make the convolutional neural network meet the expected outcomes. Without affecting the gradient descent, performing different preprocessing on the input image or performing different transformations on the output of the convolution results can enable CNN to realize other desired target, e.g. object detection at different scales.

Convolutional neural networks use BP framework for learning in supervised learning. BP in convolutional neural networks is divided into three parts, i.e., backward propagation of fully connected layer with convolutional kernel and backward pass of pooling layer^[35,52]. The BP computation of the fully connected layer is the same as that of the traditional feed-forward neural network. The forward propagation of the convolution is computed by the formula Eq (3-1).

$$\mathbf{Z}^{l+1}(i, j) = [\mathbf{Z}^l \otimes \mathbf{w}^{l+1}](i, j) + \mathbf{b} = \sum_{k=1}^{K_l} \sum_{x=1}^f \sum_{y=1}^f [\mathbf{Z}_k^l(s_0 i + x, s_0 j + y) \mathbf{w}_k^{l+1}(x, y)] + \mathbf{b}$$

$$(i, j) \in \{0, 1, \dots, L_{l+1}\} \quad L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1 \quad (3-1)$$

The summation part in the equation is equivalent to solving a cross correlation. $\mathbf{b}$ is the deviation; $\mathbf{Z}^l$ and $\mathbf{Z}^{l+1}$ represent the convolutional inputs and outputs of layer l , namely the feature map. L^{l+1} is the size of $\mathbf{Z}^{l+1}$, assuming that the feature map has the same length and width. $\mathbf{Z}(i, j)$ corresponds to the pixels of the feature map, K is the number of channels in the feature map; f , s_0 , and p are convolution layer parameters, corresponding to the convolution kernel size, convolution step size (stripe), and padding layer number.

In particularly, when the convolution kernel is size $f=1$, step $s_0=1$ and it does not contain the padding of the unit convolution kernel, the cross-correlation calculation in the convolution layer is equivalent to matrix multiplication, and thus a fully connected network is constructed between convolution layers as Eq (3-2)

$$\mathbf{Z}^{l+1} = \sum_{k=1}^{K_l} \sum_{i=1}^L \sum_{j=1}^L \left(\mathbf{Z}_{i,j,k}^l \mathbf{w}_k^{l+1} \right) + \mathbf{b} = \mathbf{w}_{l+1}^T \mathbf{Z}_{l+1} + \mathbf{b}, \quad L^{l+1} = L \quad (3-2)$$

The back propagation of convolutional layers is a cross correlation calculation similar to forward propagation:

$$\left(\frac{\partial \mathbf{E}}{\partial \mathbf{A}} \right)_{i,j}^l = \sum_{k=1}^{K_l} \sum_{x=1}^f \sum_{y=1}^f \left[\mathbf{w}_k^{l+1}(x, y) \left(\frac{\partial \mathbf{E}}{\partial \mathbf{A}} \right)_{s_0 i + x, s_0 j + y, k}^{l+1} \right] f'(\mathbf{A}_{i,j}^l)$$

$$\mathbf{w}^l = \mathbf{w}^{l+1} - \alpha \left(\frac{\partial \mathbf{E}}{\partial \mathbf{w}} \right)_k = \mathbf{w}^{l+1} - \alpha \left[\mathbf{A}^{l+1} \left(\frac{\partial \mathbf{E}}{\partial \mathbf{A}} \right)_k^{l+1} \right] \quad (3-3)$$

Where E is the error calculated by the cost function, f' is the derivative of the excitation function, α is the learning rate, and if the forward propagation of the convolution kernel is calculated by using convolution, the backward propagation also flips the convolution kernel to perform convolution. The forward and backward propagation process of convolution is shown in Fig. 3-3.

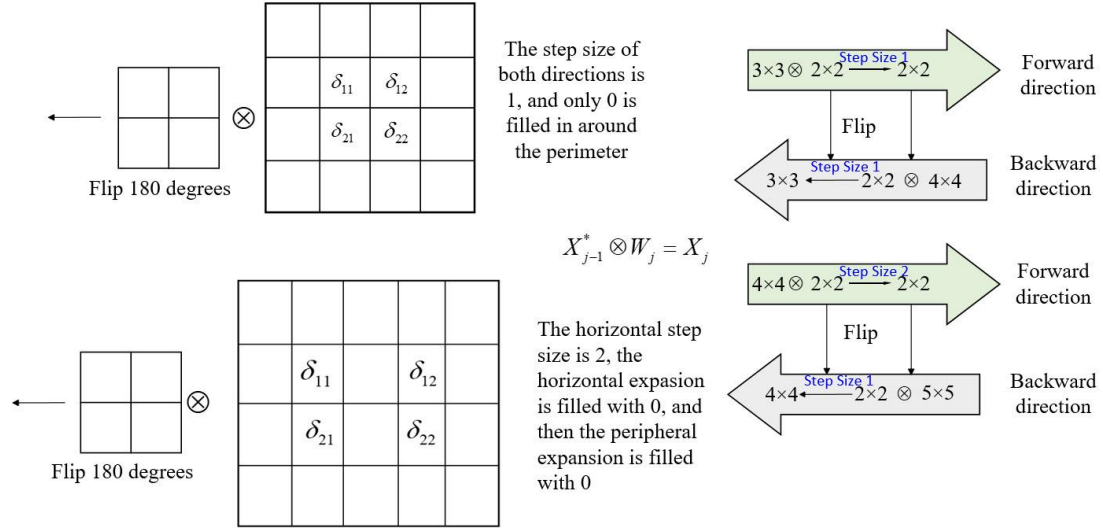


Figure 3-3 Forward and backward propagation process of convolution kernel

The pooling layer selects the pooling region in the same step as the convolutional kernel scan feature map and is controlled by the pooling size, step size and padding. Its general representation is :

$$\mathbf{A}_k^l(i, j) = \left[\sum_{x=1}^f \sum_{y=1}^f \mathbf{A}_k^l(s_0 i + x, s_0 j + y)^p \right]^{\frac{1}{p}} \quad (3-4)$$

where the step size s_0 , pixel (i, j) has the same meaning as convolutional layer, and p is a pre-specified parameter. When $p = 1$, the pooling takes the mean value in the pooling region and is called average pooling; when $p \rightarrow \infty$, the pooling takes a great value in the region and is called max pooling. The pooling layer has no parameter updates in the back propagation, so only the errors need to be assigned to the appropriate locations in the feature map according to the pooling method. For max pooling, all errors are assigned to the location of the maximal value; for average pooling, the errors are equally distributed throughout the pooling region^[52]. The forward and backward propagation process of pooling is shown in Fig. 3-4.

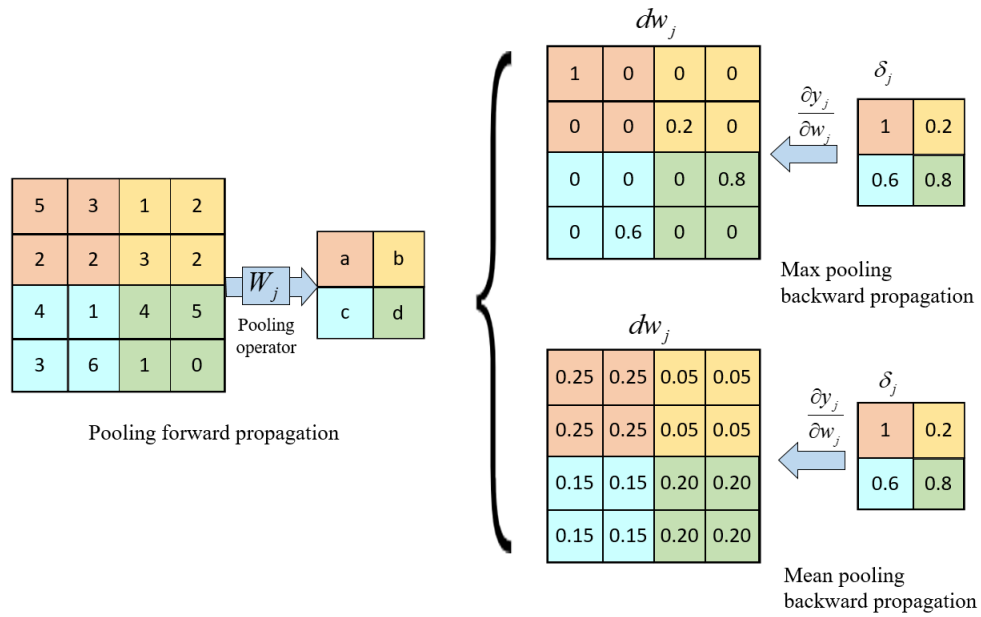


Figure 3-4 Forward and backward propagation process of pooling

3.1.3 Loss functions

Gradient back propagation solves the optimization process of network model covariates with the objective of minimizing the classification or regression loss function. In the iterative solution process of using gradient descent, factors such as unbalanced data categories or coupling between multiple tasks may cause a loss to prevail in training and lead to training failure. Therefore, the selection of an appropriate loss function and the adoption of necessary measures to eliminate the influence of unfavorable factors are crucial for network training.

In the image processing task, regression usually refers to the border regression in the object detection task, and the applied loss function is called as regression loss function. Classification refers to the region classification in the object detection task or refers to the pixel classification in the semantic segmentation task. The utilized loss function is called as the classification loss function. For the multi-image processing task that includes classification and regression, the weighted sum of multiple loss functions of regression and classification is called as the compound loss function. There are many different loss function expressions, among which mean squared error (MSE, also known as L_2 loss) is most used in regression and cross-entropy loss is more common in classification.

The task of deep learning is essentially to fit a function to map the input image to the corresponding label. The quality of this mapping needs to be expressed by a loss

function, and the process of learning and training is also guided by the loss function. This section summarizes some common loss functions in convolutional neural networks, their characteristics and usage occasions.

There are two common tasks of convolutional neural networks; one is classification (output discrete values, such as predicting the category of input pictures), and the other is regression (output continuous values, such as predicting the relative offset of the regression box). Both tasks have their own suitable loss function, sometimes a combination of multiple loss functions. For convenience, the respective loss functions are discussed separately based on different tasks.

3.1.3.1 Cross entropy loss function

Cross-entropy is commonly used in machine learning as a loss function. As a most important cost function, it is used to optimize classification models like logistic regression and artificial neural networks. Cross-entropy is a measure from the field of information theory, building upon entropy and generally calculating the difference between two probability distributions. It is closely related to KL divergence (Kullback–Leibler divergence) that calculates the relative entropy between two probability distributions, whereas cross-entropy is to calculate the total entropy between the distributions.

Though cross-entropy is different from KL divergence, but it can be calculated by using KL divergence. It is different from log loss, but it calculates the same quantity when used as a loss function. So it is used to evaluate the difference between the probability distribution of the Softmax output and the true probability of the training sample. Moreover, from an optimization point of view, the cross-entropy loss is suitable for Softmax. Because of the concise meaning, cross-entropy loss function is applied in this dissertation chapter 5.

3.1.3.2 Hinge loss function

The hinge loss function $L = \max(0, 1-x)$, whose function curve is shown in the blue line in Fig. 3-5, is named due to its similarity to a hinge. The hinge loss function originated from SVM (Support Vector Machine), which is characterized by correct and wrong classification points. Considering its distance from the classification plane, the greater the distance is, the greater the loss will be. But for samples that can be

correctly classified, if these samples are far from the classification plane, or if the distance is not large enough, there will be a little loss. This forces the optimization process to find the maximum interval to divide plane during optimization.

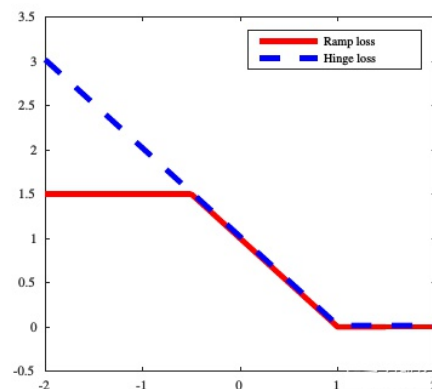


Figure 3-5 Loss function illustration

3.1.3.3 Central loss function

The loss function introduced above only considers the features that have been learned to increase the interval between different classes. Good features should not only consider that the interval between different classes is large, but also ensure that the same class. Smaller intervals make such features more discriminative. The central loss function can guide the features learned by the network to gather together as much as possible when they are of the same kind.

$$L_{center\ loss} = \frac{1}{2} \sum_{i=1}^N \|x_i - c_{yi}\|_2^2 \quad (3-5)$$

The above formula Eq. (3-5) is the expression of the center loss function, where c_{yi} is the mean point of the depth feature of the y_i -th training sample. The farther the network learns the features from the center point, the greater the penalty, which promotes the network to learn the features that allow each sample of the same category to gather closer.

In actual application, since the value of the center point loss function considers the differences within categories, and the cross-entropy loss function only considers the differences between categories, the center loss function and the cross-entropy loss function are generally combined to obtain the desired value. In this way, the final loss function of the network can be expressed as:

$$L_{final} = L_{cross\ entropy\ loss} + \lambda L_{center\ loss}(h, y_i) \quad (3-6)$$

Among them, λ represents the relative proportion of the two losses, and is used to adjust the relative strength between the learned feature categories and the clustering within the category. The larger the λ is, the stronger the clustering within the category will be.

In terms of classification performance, the features learned by the center loss function combined with the traditional cross-entropy loss function are better than those learned by only using the traditional cross-entropy loss function.

3.1.3.4 Smooth-L1 loss function

Smooth L_1 loss was first proposed in the Fast RCNN algorithm as the loss function of Bounding Box Regression. It is an improved version of L_1 loss. Comparing with ordinary L_1 loss and L_2 loss, the following three advantages can be observed.

- (1) In the later stage of training, when the difference between the predicted value and the true value is small, the derivative of Smooth L_1 is smaller than that of L_1 , which can make the training achieve higher accuracy.
- (2) In the early stage of training, when the difference between the predicted value and the true value is large, the derivative of Smooth L_1 will not be too large compared with the derivative of L_2 , which avoids the situation of unstable training.
- (3) For data noise and outliers, the loss of Smooth L_1 is smaller than that of L_2 , and it is not easy to mislead the network training. In other words, it is more robust to noise and outliers.

Based on the above advantages, after Fast R-CNN (Region based CNN), many popular target detection algorithms, such as Faster R-CNN and SSD (Single Shot Detector), also use the loss function of Smooth L_1 .

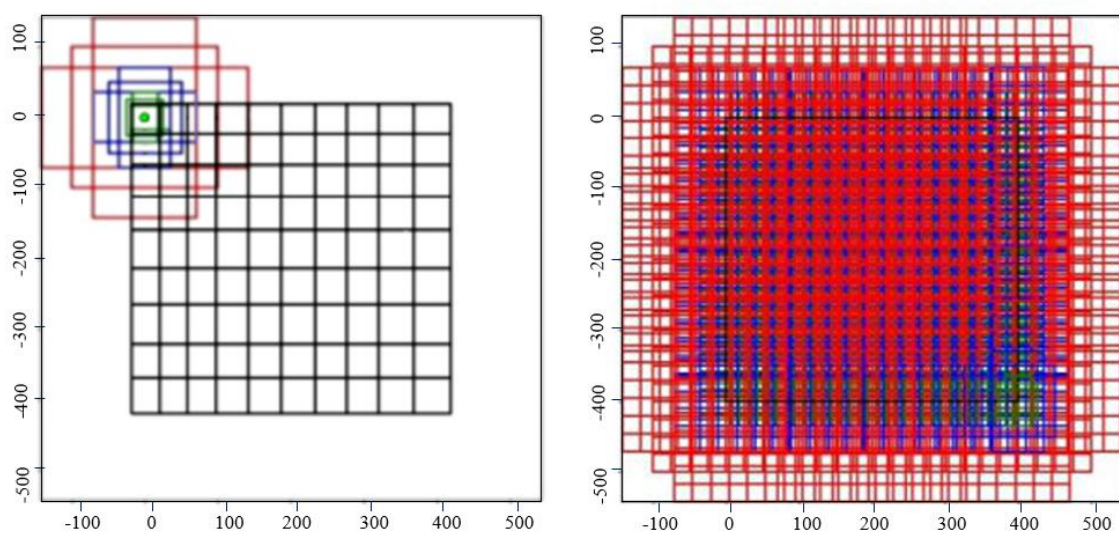
Summary

When working on the deep learning problem, loss functions are used to optimize the model during training. The objective is almost always to minimize the loss function. The lower the loss, the better the model. According to the classification and regression problems of deep learning, several common loss functions are introduced above. Some of these loss functions can be used interchangeably (for example, the L_2 loss function is also often used in classification problems). The loss function needs to be constructed or selected according to the specific problem to be solved. The general idea is to guide

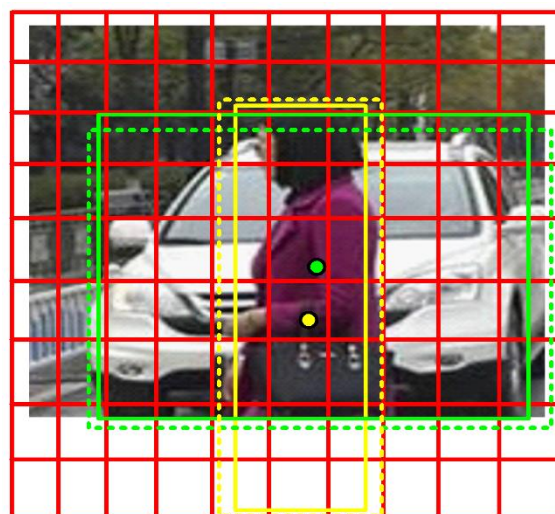
the network to learn features with strong discrimination (large inter-class distance and small intra-class distance), and easy to be optimized.

3.1.4 CNN basic framework

CNN should ensure feature invariance. The object size and position are random in the image, and CNN has abandoned the traditional sliding window localization method in object localization, and replaced it with the predefined Anchor boxes method, which is proposed in the RPN of Faster RCNN. The so-called predefined anchor boxes are empirically set up with several rectangles of different scales and shapes, with large



(a) The center of the anchor frame on the feature map (b) The arrangement of the anchor frame on the image



(c) Anchor box (solid box) with target tag position (dashed box) for locking the target

Figure 3-6 Anchor box for region candidate process

scales corresponding to high level low-resolution feature maps and small scales corresponding to low level high-resolution feature maps. The number of rectangles is the same as the number of feature image pixels, the center point is aligned with the pixels of the feature map, and the rectangles at each scale have multiple width-to-height ratios, representing different shapes at that scale. That is, there may be different shapes of objects at each feature image pixel position, such as thin high shapes or flat shapes, etc., as in Fig. 3-6 (a). All anchor frames at each scale are mapped to the input image and form a dense anchor frame arrangement, as in Fig. 3-6(b). Among these anchor frames, there must be the one with the largest overlap area with the target object label position, as in Fig. 3-6 (c). The position of this anchor box is the target object area candidate, and the predicted position of the target object is regressed by supervised learning from the anchor box position to the label position.

As mentioned above, CNN naturally has a pyramid feature distribution characteristic from low to high resolution. Combined with the characteristics of anchor frame localization, it is easy to form the CNN framework as shown in Fig. 3-7. The features of the last layer are used for prediction in Fig. 3-7 (a). The input is the image pyramid, and the scale change is completed by the image pyramid. Different scale features of different layers are used for prediction as shown in Fig. 3-7 (b). The top-level and low-level features are fused and predicted separately as shown in Fig. 3-7 (c). Both frameworks (a) and (b) are single-feature prediction, and the single-layer feature only reflects a certain aspect of the target object attribute. With the increase of the target object categories, the invariance becomes worse, so feature fusion is an effective method to improve the performance of CNN.

The features calculated by the deep convolutional network show a hierarchical distribution, and the application of the down-sampling layer makes the feature distribution become a multi-scale pyramid shape. This feature pyramid has the characteristics of different resolutions at different scales. Previous studies have shown that the feature map of CNN from the bottom to the top shows a gradual transition from extracting low-level features such as point and color of the image to extracting edge and texture features of the image, and then extracting high-level features such as local shape and global shape of the target object. The low-level features have less semantic information, but the target object position is accurate and the resolution is high. The high-level features have rich semantic information, but the target object position is

rough and the resolution is low. By fusing multi-scale information, objects of different sizes are predicted, and the performance of the model is greatly improved.

The feature fusion of CNN can be attributed to the “pyramid extension” technology, which can be summarized into two implementation methods, which are called “hierarchical pyramid” and “scale pyramid” respectively, as shown in Fig. 3-8. The hierarchical pyramid stacks the features of the upper layer by up-sampling and then connects them with the lower layer features, and finally realizes the fusion of feature vectors at different levels and scales. The scale pyramid up samples the features of a certain layer in different proportions, and then stacks the features by up sampling to the same size, and finally realizes the feature fusion of the same level and different scales. Under the general idea of pyramid extension, a large number of correlation structures^[53,54,55] have emerged, and these methods have made great improvements in calculation speed and accuracy.

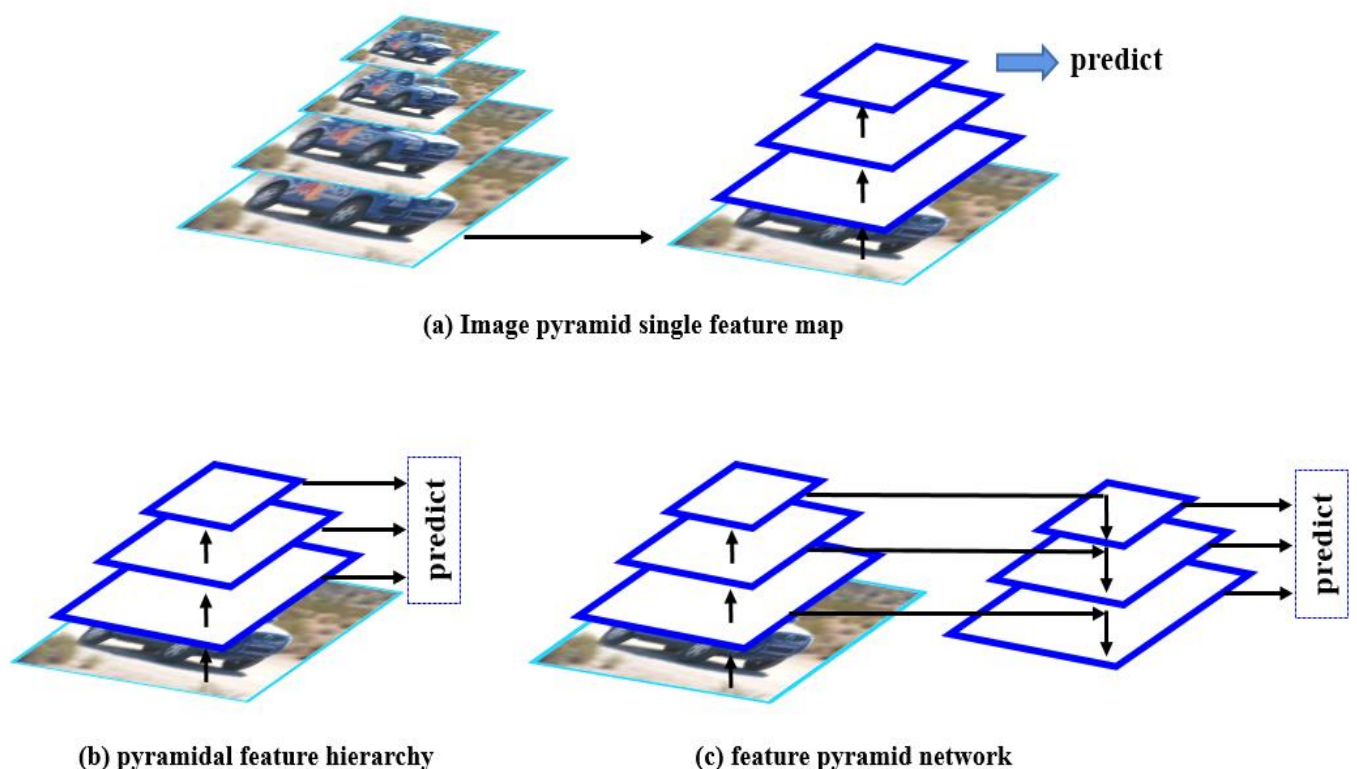
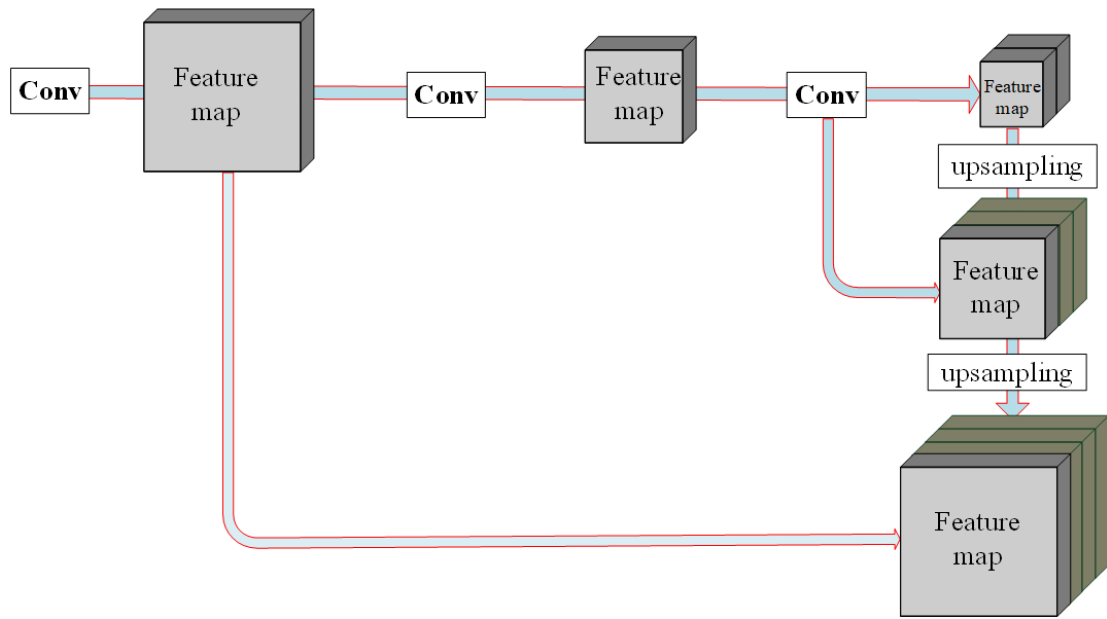
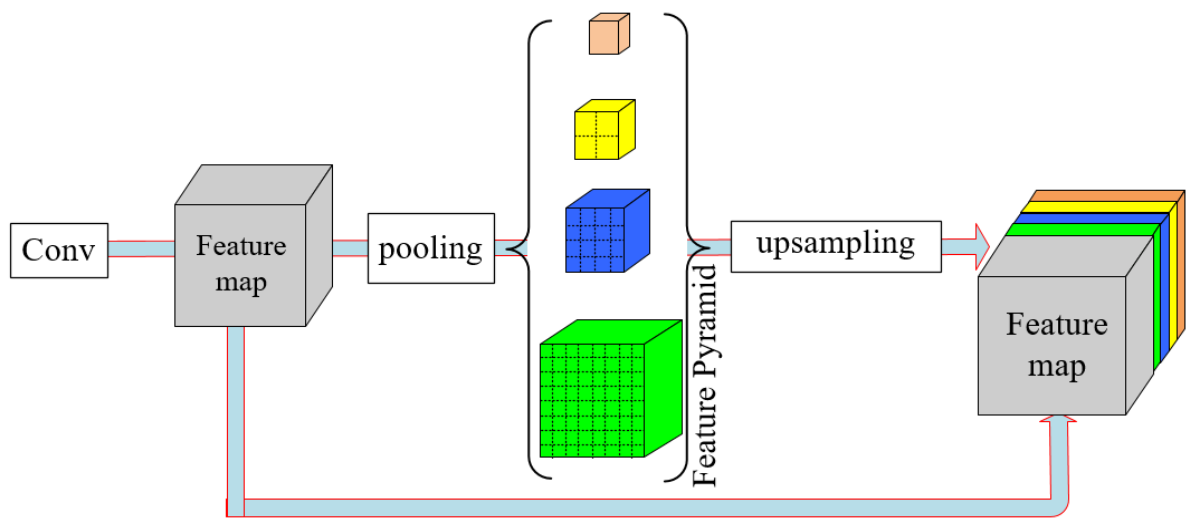


Figure 3-7 Common CNN frameworks



(a) Hierarchical Pyramid



(b) Scale Pyramid

Figure 3-8 Structure of feature pyramid

3.2 Principal component analysis (PCA)

3.2.1 PCA introduction

Principal Component Analysis (PCA) ^[17] is classified as one of the top ten algorithms for machine learning. It is used to identify a small number of representative principal components of features from a larger set of data. Principal Component Analysis is a tool invented by Karl Pearson^[56] in 1901 for predictive modeling and seeking of data analysis. Principal component analysis is a useful statistical method that is now widely used in areas such as image compression, face recognition, neuroscience, and image processing. Unexpectedly, the algorithm now becomes so popular after 120 years.

PCA makes data easier to explore and visualize. It is a simple non-parametric technique for extracting information from complex and messy datasets. Now is the era of big data, and this algorithm is undoubtedly widely used. The point of PCA is to get the maximum of variance with the fewest principal components. Data compression is also supported once patterns have been discovered in related data. People use PCA to eliminate the number of variables, or when there are too many predictors compared to the number of observations, or to avoid multi-collinearity. It is closely related to canonical correlation analysis, which utilizes an orthogonal transformation to convert a set of observations containing correlated variables into a set of values called principal components. The number of principal components used in PCA is less than or equal to the smaller number of observations. Principal component analysis is sensitive to the relative scale of the original variables. PCA is widely used in many fields such as market research, social sciences, and industries that use large datasets. The technique can also help provide a low-dimensional picture of the raw data. PCA can reduce complex and confusing datasets to a useful set of information with minimal effort.

PCA belongs to linear transforms (Karhunen-Loeve or Hotelling transform) based on the statistical techniques. This method provides a powerful tool for data analysis and pattern recognition which is often used in signal and image processing^[57,58] as a technique for data compression, data dimension reduction or their decorrelation as well. There are various algorithms based on multivariate analysis or neural networks^[59,60] that can perform PCA on a given data set. Presented paper introduces PCA as a possible tool in image enhancement and analysis.

Principal component analysis, also known as the Karhunen Loeve Transform, is a technique used to explore high-dimensional data structures. PCA is commonly used for exploring and visualizing high-dimensional datasets. It can also be used for data compression, data preprocessing, etc. PCA can synthesize high-dimensional variables that may have correlations into linearly unrelated low-dimensional variables, known as principal components. The new low dimensional dataset preserves the variables of the original data as much as possible.

3.2.2 The basic principle of PCA

Linear PCA is implemented by means of a linear transformation (matrix). Mathematically, the eigenvector of a linear transformation whose direction is invariant under that transformation and the proportion of that vector scaled under this transformation is called the eigenvalue. A linear transformation can usually be fully described by its eigenvalues and eigenvectors. In linear algebra, the dot product between two vectors is considered as the projection of column vectors to row vectors, then matrix multiplication can be considered as the projection of each column vector in the right matrix to the space represented by the base of each row vector in the left matrix. Choosing different bases transformations matrices can give different representations of the same dataset. From the perspective of information preservation, the goal of PCA is to choose m bases so that the projected data retains the original information to the maximum extent. Information can be measured by information entropy. If the sample satisfies Gaussian distribution, the information entropy is proportional to the variance, i.e., the larger the variance, the greater the variation of the data and the more information it can contain. Thus, the goal of PCA is to maximize the variance of the data after dimensionality reduction.

Given n observations $x_1, x_2, \dots, x_n \in \mathbb{R}^p$, form a matrix X , $X \in \mathbb{R}^{p \times n}$, i.e.,

$$X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1n} \\ X_{21} & X_{22} & \dots & X_{2n} \\ \vdots & \vdots & \dots & \vdots \\ X_{p1} & X_{p2} & \dots & X_{pn} \end{bmatrix} \quad (3-7)$$

Our goal is to convert such a set of p -dimensional data into m -dimensional data, that is, we require a $p \times m$ basis transformation matrix V that converts the original X matrix into a Z matrix:

$$Z_{m \times n} = V_{p \times m}^T X_{p \times n}, \text{ where } V \in \mathbb{R}^{p \times m}.$$

$$V_i = \begin{bmatrix} V_{1i} \\ V_{2i} \\ \vdots \\ V_{pi} \end{bmatrix} \quad (3-8)$$

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{p1} & v_{p2} & \cdots & v_{pm} \end{bmatrix} = [v_1 \quad v_2 \quad \cdots \quad v_m] \quad (3-9)$$

$$V^T = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{p1} & v_{p2} & \cdots & v_{pm} \end{bmatrix} = \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{bmatrix} \quad (3-10)$$

Let the covariance matrix corresponding to the original data matrix X be C . Let the covariance matrix of Z be D . Then we have:

$$\begin{aligned} D &= \frac{1}{n} Z Z^T \\ &= \frac{1}{n} (V^T X) (V^T X)^T \\ &= \frac{1}{n} V^T X X^T V \\ &= V^T \left(\frac{1}{n} X X^T \right) V \\ &= V^T C V \end{aligned} \quad (3-11)$$

where $C = \text{var}(X) = \frac{1}{n} X X^T$.

The problem translates into:

$$\min_v \text{tr}(V^T C V), \text{ s.t. } V^T V = I \quad (3-12)$$

where tr is the trace of the matrix.

The Lagrange multiplier method is applied to the solution, the multiplier matrix $\Theta \in \mathbb{R}^{m \times m}$, considering the constraint $V^T V = I$, the Lagrange multiplier matrix is diagonal

array at this time, i.e., $\Theta = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$, which translates into solving for the extrema of the Lagrangian function as follows:

$$L(V, \lambda) = \text{tr}(V^T C V) + \text{tr}(\Theta^T (V^T V - I)) \quad (3-13)$$

The partial derivative of V takes its extreme value at the derivative of Θ :

$$\begin{aligned} \frac{\partial L(V, \lambda)}{\partial V} &= -2CV + 2V\Theta \\ \Rightarrow CV &= V\Theta \\ \Rightarrow Cv_i &= \lambda_i v_i, i = 1, 2, \dots, m \end{aligned} \quad (3-14)$$

According to the definition of eigenvectors, the above equation λ_i, v_i represents the eigenvalues and eigenvectors of the covariance matrix C , respectively. The larger the eigenvalues, the richer the information contained in the data in the direction of its corresponding eigenvectors. Obviously, at this point, it is only necessary to make $\lambda_1, \lambda_2, \dots, \lambda_m$ and $v_1, v_2, \dots, v_m$ be the first m largest eigenvalues and unit eigenvectors of the covariance C , respectively, to make the objective function reach the optimal value.

From the above derivation, PCA method assumes that the function mapping from high-dimensional space to low-dimensional space is linear and is achieved by linear combination of the original features, and it is difficult to use linear PCA for dimensionality reduction when the original data cannot be represented as a linear combination of features. In addition, PCA method assumes that the input data conforms to a Gaussian distribution, and the principal elements derived in the case of non-Gaussian distribution may not be optimal. Thus, there are many extensions to basic PCA that can address its drawbacks, such as robust PCA^[61], and 2-D PCA.

As discussed above, the process of solving the principal components of a data sample is the process of solving the eigenvalues and eigenvectors of its covariance matrix.

3.2.3 PCA usage for image compression

Data volume reduction is a common task in image processing. There are a lot of algorithms^[34,35,62] based on various principles leading to the image compression. Algorithms based on the image color reduction are mostly that image data is thrown away as part of the compression process but their results are still acceptable for some real world applications. The image transformation from color to the gray-level

(intensity) image belongs to the most common algorithms. Its implementation is usually based on the weighted sum of three color components R, G, B.

The RGB matrices contain image color components, and the weights are determined with regards to the subjective human perception [35].

The process of image dimensionality reduction using PCA directly has a very wide range of applications in video storage and dissemination, and the principle of its implementation is very simple. For a RGB color image with resolution n , each pixel is regarded as a sample, and each sample is represented by a 3-dimensional vector composed of R, G, and B values, then the image can be expanded into a $3 \times n$ -dimensional matrix X (here the pixels of each channel of the image are connected first and last by column, and the length is n), where 3 denotes the number of dimensions and n denotes the number of samples. Based on the matrix X , dimensionality reduction can be performed by the above PCA principle. Since the data is 3-dimensional, a 3×3 covariance matrix can be obtained, corresponding to 3 eigenvalues and 3-dimensional eigenvectors. The resulting covariance array, eigenvalues and eigenvectors are shown in Table 3-1.

As seen in Fig. 3-9, the image (c) reconstructed based on the first two principal components of PCA is already very close to the original image (a) in terms of color, and the difference is basically not perceived by the naked eye, but the size of the image has been reduced from 236k to 100k.

Table 3-1 PCA processing results

	1	2	3		1	2	3		1	2	3
1	0.0707	0.0702	0.0714	1	0.2149	0	0	1	0.5705	0.8112	-0.1284
2	0.0702	0.0712	0.0725	2	0	0.0012	0	2	0.5749	-0.2828	0.7678
3	0.0714	0.0725	0.0743	3	0	0	2.0290e-04	3	0.5866	-0.5118	-0.6277

(a) Covariance array

(b) Eigenvalues

(c) Eigenvectors



Figure 3-9 Original and reconstructed image size

3.2.4 PCA usage for feature determination

Properties of PCA can be used to extract the features for the determination of selected object orientation or its rotation, object at different scale, and blurred object as well as noise contaminated object^[62,63]. Various method of image segmentation to object definition (like threshold, edge detection^[18]) must be used at first. Binarized image containing object boundary or its area in black (or white) pixels on the inverse background results from this process. After that two vectors a and b containing the Cartesian x and y coordinates of object pixels can be simply formed.

Here the calculation formulation is omitted as the theoretic description can be referred to Wikipedia website.

The very similar normalized eigenvalues computed by PCA can be obtained as shown in Table 3-2. The input images are including the object at large scale (a), small scale (b), blurred (c) and noisy (d) as well as rotated (e). Intuitively, the normalized PCA eigenvalues can be used as the feature fusion into the deep CNN structure according to the detection purpose of the target object. Based on the practical calculation results, the obvious consistent eigenvalues (which are normalized) are demonstrated: $n\lambda_1 \approx n\lambda_2 \approx n\lambda_3 \approx n\lambda_4 \approx n\lambda_5$. This fact tells that the same object at different scale and rotation conditions, still retains the domain eigen-characteristics.

Table 3-2 Featurized eigenvalues of PCA

Normalized Eigenvalue:			
Figure	No.	1	2
(a)	$n\lambda_1 \approx$	0.9359	0.3524
(b)	$n\lambda_2 \approx$	0.9405	0.3397
(c)	$n\lambda_3 \approx$	0.9420	0.3357
(d)	$n\lambda_4 \approx$	0.9362	0.3513
(e)	$n\lambda_5 \approx$	0.9502	0.3117

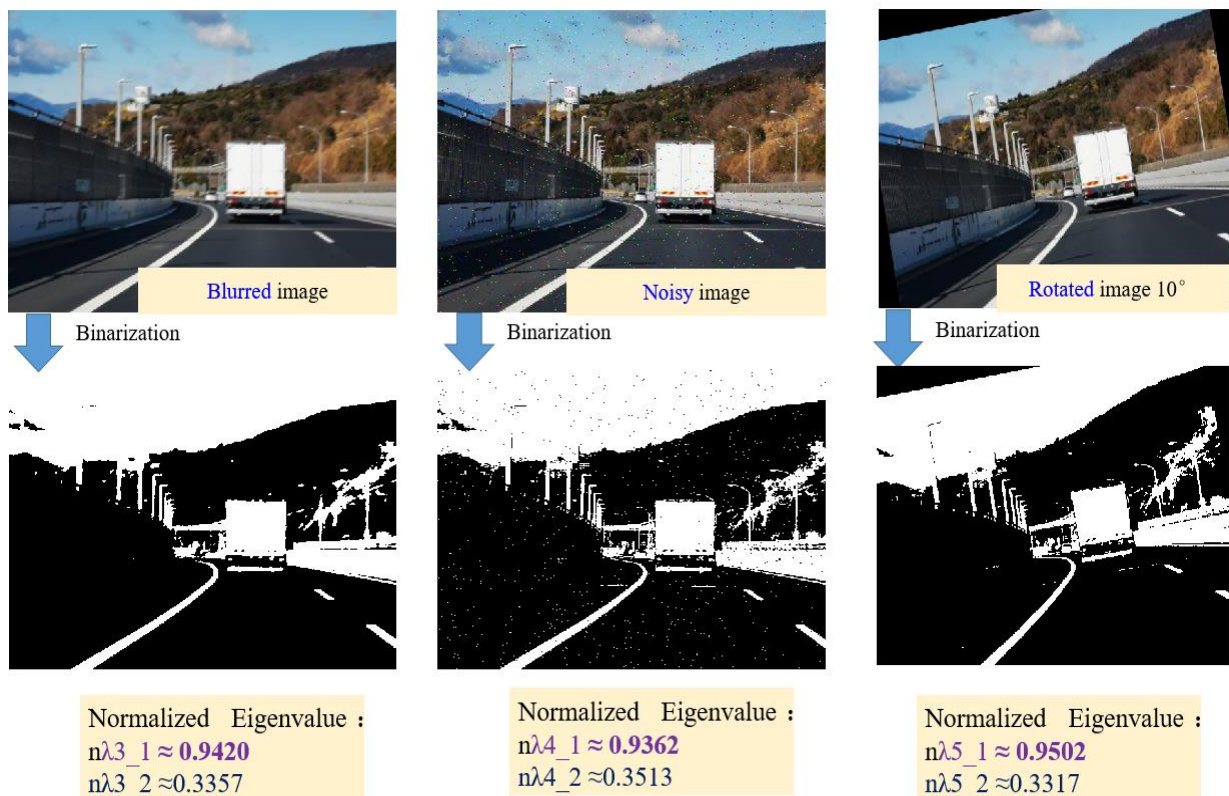
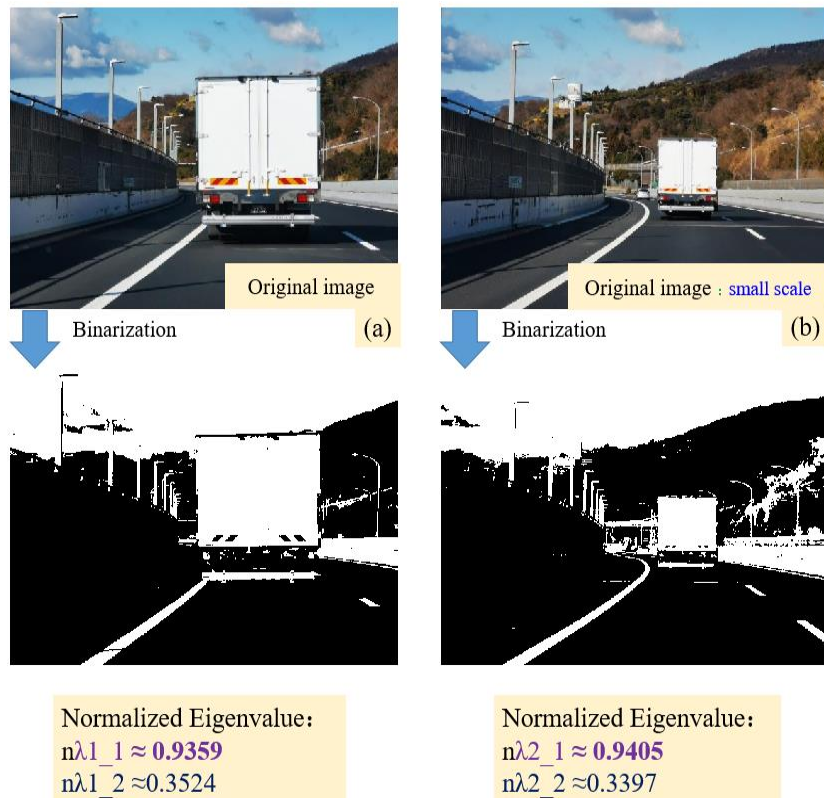


Figure 3-10 Feature of PCA eigenvalues

3.3 Discrete cosine transform (DCT)

In 1974, Ahmed and Rao proposed to use discrete cosine transform (DCT) ^[63] for image compression coding. After decades of development, DCT coding has laid a solid theoretical and practical foundation for image and video data compression with high compression ratio and high signal-to-noise ratio.

Like Fast Fourier Transform (FFT), DCT is an important transformation method with both theory and practical application. DCT is a common signal processing transformation, which quantizes the signal based on the limited signal frequency band, and converts the limited components into real values. DCT reduces the dimension of the new data matrix, and then obtains the original signal through the inverse cosine transform. It can convert the signal in the spatial domain to the frequency domain, and has good decorrelation performance. DCT transform itself is lossless, so it is favorable in the image coding area. Because discrete cosine transform is symmetric, its inverse transform can be used after quantization and coding to restore the original image information at the receiving end.

In image processing, DCT can be used to extract features, such as extracting useful feature pixel blocks, extracting image texture, and further processing such as image segmentation. DCT can also be used to speed up the transfer rate, thereby increasing processing speed. In addition, it can also be used to improve robustness of the image signals to noise. In short, DCT has strong anti-noise performance and high efficiency in extracting image information. It is an important image processing method and is indispensable in many image processing applications.

One-dimensional DCT transformation is the basis of 2-D DCT transformation. There are many forms of one-dimensional DCT transformation, among which the most commonly used is the below formulation due to its simple operation and wide applications. The definition of one-dimensional DCT is as follows:

Let a real sequence of N points $\{f(x) \mid x = 0, 1, \dots, N-1\}$, whose DCT and inverse DCT are defined as:

$$F(u) = C(u) \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)u\pi}{2N} \quad u=0,1,\dots,N-1 \quad (3-15)$$

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u) F(u) \cos \frac{(2x+1)u\pi}{2N} \quad x=0,1,\dots,N-1 \quad (3-16)$$

Among them,

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & \text{else} \end{cases}$$

If $[C]$ is defined as a transformation matrix and $[f(x)]$ is a spatial data array.

If $[F(u)]$ is an array of transformation coefficients, then one-dimensional DCT can be written as follows:

$$[F(u)] = [C][f(x)] \quad [f(x)] = [C]^T [F(x)]$$

The definition of 2-D DCT is as follows:

Based on one-dimensional DCT, 2-D DCT can be derived.

Let $\{f(x, y) \mid x=0,1,\dots, M-1; y=0,1,\dots, N-1\}$ be the set of 2-D image signal sequences, whose forward and inverse transformations are defined as:

$$F(u, v) = \frac{2}{\sqrt{MN}} C(u)C(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2M} \cos \frac{(2y+1)v\pi}{2N} \quad (3-17)$$

$$f(u, v) = \frac{2}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} C(u)C(v) F(u, v) \cos \frac{(2x+1)u\pi}{2M} \cos \frac{(2y+1)v\pi}{2N} \quad (3-18)$$

Among them, $u=0, 1, 2, \dots, M-1; v=0, 1, 2, \dots, N-1$; The definitions of $C(u)$ and $C(v)$ are the same as the previous one-dimensional DCT.

If $[C]$ is defined as a transformation matrix, $[f(x, y)]$ is a spatial data array.

If $[F(u, v)]$ is an array of transformation coefficients, the 2-D DCT can be written as follows:

$$[F(u, v)] = [C][f(x, y)][C]^T \quad [f(x, y)] = [C]^T [F(x, y)][C]$$

Forward Discrete Cosine Transform (FDCT, Forward DCT) transforms data in the spatial domain into coefficients in the frequency domain, which is reversible. The inverse discrete cosine transform (IDCT, Inverse DCT) restores the coefficients in the frequency domain to pixels in the spatial domain during decoding. In video coding, 2-D transformations are performed to remove data correlation and gather energy. The first coefficient $F(0,0)$ in the upper left corner of the matrix corresponds to the average brightness of the image $f(x, y)$, which is called the DC coefficient of the image energy; the other coefficients in the lower right corner correspond to the high-frequency part of the image $f(x, y)$, which is called the AC coefficient, i.e., the image details; from left to right represents the direction of increasing horizontal spatial frequency, and from top to bottom represents the direction of increasing vertical spatial frequency. The

decorrelation and compression effect of discrete cosine transform will increase with the size of the selected image block, but its computational complexity will also increase exponentially.

In this thesis study, DCT is used for edge line density extraction; the detail including visual pictures can be found in chapter 4.

3.4 Summary

The theoretic base for proceeding the research is described in order to have an overview. The general conceptual techniques of convolutional neural network, loss function, principal component analysis, and discrete cosine transform are briefly introduced for reference when the comprehensive perception becomes necessary. The normalized eigenvalues calculated by principal component analysis process, are proved to be identical theoretically and the practical image process can also result in the highly consistent values. So the normalized eigenvalues can be taken as the feature of scale invariance, and can be used as the feature fusion into the deep CNN structure according to the detection purpose of the target object.

Chapter 4. Object detection from monocular camera images with depth cues extracted by CNN

4.1 Introduction

Advanced Driver Assistance Systems (ADAS) increasingly act and interact in the real world with complex environments to support driving tasks. For this purpose, more and more complex environmental detection techniques are developed with multiple sensors. These sensors can include sonar, radar, LiDAR and cameras. To implement a proper ADAS function, the functional system is often based on some models which are combined with the sensor signal inputs.

The sensor technology provides the basic external information to input to the functional module system. In the vehicles, imaging data and ranging data like distance & speed are collected through the sensors. The reactions to the objects are appropriately calculated with the determined model algorithm. Due to the complexity of vehicle dynamics and ego vehicle environment, the determined detection algorithm is very often adaptively calibrated conclusion. And the object detection algorithm is often insufficient as per the overwhelming complexity of real world. The algorithm functional insufficiencies can lead to unintended system behaviors. To make the detection algorithm more robust, currently the artificial intelligent approaches, as mentioned in reference [64], are popular and they are widely applied to conduct the processing of the deep learning and optimization. The typical deep learning models include convolutional neural network (CNN), recurrent neural networks (RNNs), and deep belief networks (DBNs).

In the area of ADAS, the low-cost monocular camera is largely used to provide the driving assistance functions. Such a camera sensor can accomplish many tasks, including the detections of lane mark, vehicles, and pedestrians as well as the distance from the ego vehicle to the obstacles. The image data returned by a monocular camera sensor can be used to issue lane departure warnings, blind spot obstacle warning, collision warnings, or to design a lane keep assist control system. In conjunction with other sensors, it can also be used to implement an emergency braking system and other critical safety feature systems.

In the real world, there are many fatal accidents due to the blind spot ignorance, especially in the case of the long truck or bus turning around the traffic corner. In Europe and Japan, the car rear view and side view mirror can be replaced by camera-

monitoring system. Therefore, the commercial mirror-less cars can be expected soon and the applied ADAS functions based on camera image processing in such monitoring system are becoming more important.

In this study, we focus on the task of vehicle blind spot detection (BSD) by monocular camera image processing. We proposed an image processing algorithm to detect the obstacle depth information in the rear side area shown in Fig. 4-1 where it is the blind spot area for the driver. The image data is captured by a monocular camera from the rear side view. Usually, whether a car is getting near to the blind spot zone or not is only judged by 2-D image information. E.g., in reference study [66], the classifier determines whether an image region is a vehicle, or a non-vehicle based on the feature vector. Paper [67] uses the vehicle shadow or light location information in the image to conduct the blind spot detection.

In this study we utilize the inferred depth information from the 2-D image and relative position to achieve the robust BSD performance augmentation. The depth information is the key to recognize the near and far area^[67,68]. In computer vision world, the depth mapping from a single 2-D image is still a challenge topic^[69]. The computer can hardly obtain precise distance value information if there is no other reference parameter in a single 2-D image. The mentioned depth information in the 2-D image is a relative scale other than absolute value. The methods of depth information estimation from images rely on the structure from motion, binocular and multi-view stereo.

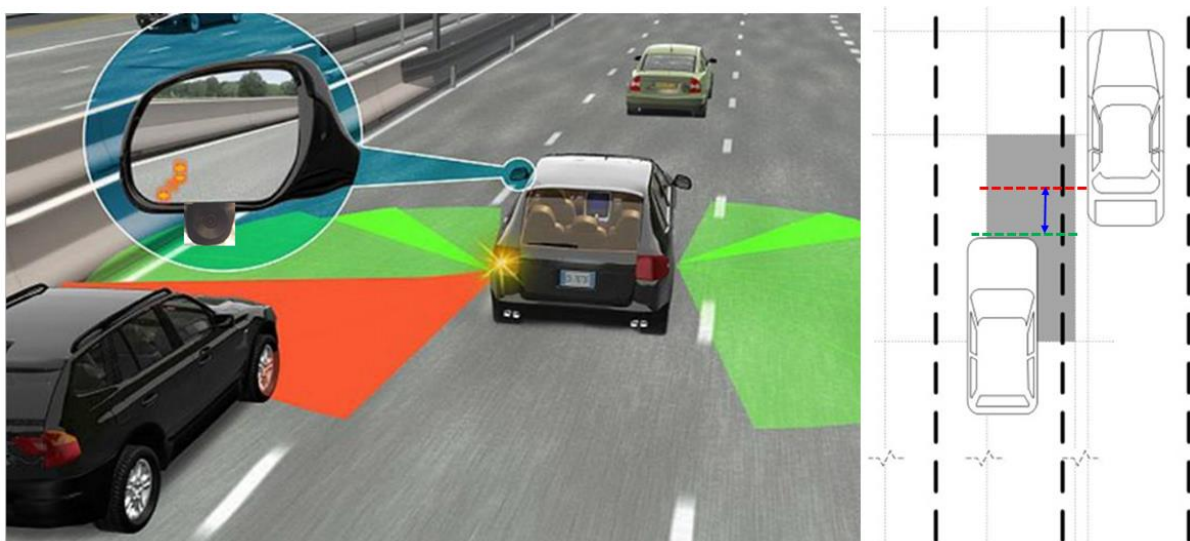


Figure 4-1 Driver blind area and side view camera

These observations come from the multiple view of the scene under different lighting conditions. To overcome these limited conditions, the monocular depth estimation as a supervised learning task attempts to directly predict the depth of each pixel in an image by the offline trained model. The learning-based methods have proved effective for the depth estimation in single images. Paper [70] proposed a CNN training objective to learn to perform single image depth estimation with an image reconstruction loss by the generated disparity images. However, this approach uses binocular stereo footage to enforce consistency between the disparities produced relative to both the left and right images.

In this study, the monocular depth cue plays the key rule. Supposed that the monocular camera focus is set at an infinite point or far spot, the depth information is inferred from the comparisons of the blur cues^[8] and texture cues^[5,6,7]. The blur cues and the texture cues are extracted from the image. The texture cues mean the density of the edge inside the block, while the blur cues mean the degree of blur and the sharpness of the edge. We proposed to conduct the local cues extraction from the monocular image. The depth estimation is derived from the feature comparison between two image blocks selected within a single image. In our proposed algorithm, the following methods are used.

Coarse-to-fine analysis is used to increase the probability of feature extraction from the limited image block area, because the interested information of the large-scale feature may get lost in the high-resolution image block, but it is included in the lower resolution image block. The low-resolution level is ideal to get an overview of the image scene. While the details can be found at higher or fine resolution level^[65].

Principal component analysis (PCA) is applied to extract the most important information from the image data^[18]. As we know, PCA is one of the statistical techniques used for the image data compression and data structure analysis. It is used to extract the edge lines as the texture cues from a single image. In order to classify the edge line orientations for higher spatial frequency density, we proposed to separate the PCA result into four categories or zones. This is for the purpose of obtaining more clear texture cues with the higher spatial frequency density along the edge line orientation.

Discrete cosine transform (DCT) is applied to extract the features of texture cues and blur cues from the image. DCT detects the edge line density allocated in spatial

frequency domain with regarding to the texture cues. But for the blur cues, DCT detects the sharpness of the edges.

CNN is applied to classify the depth cues with the input of DCT analysis results. Through deep learning, a better trained network model with acceptable accuracy is obtained for the experiments.

The effectiveness of the proposed approach by using the above method combination is testified with some case studies^[13,14]. In this chapter, the outline of the proposed approach is described, and one of the test results is shown. The application for BSD purpose is conducted with some real road traffic image data.

As the conclusion of this work, the proposed method is effective to detect the depth cue information for the blind spot detection purpose. The adaptation to more complex environment and the algorithm performance improvement are planned as per the study progress.

4.2 Algorithm

In this session, the involved basic theory and methods are briefly introduced. Both of the texture cue and blur cue are used to obtain the depth information, other than the exiting methods that only one of the cues is applied^[7]. Depth perception allows us to perceive the world around us in three dimensions and to estimate the distance of object from ourselves and from other objects. One way of the depth perception is the use of monocular cues. When perceiving the world around us, many of these monocular cues work together to contribute to our experience of depth estimation. In computer vision, the image taken by monocular camera is what the computer sees. When the monocular camera makes the focus at an object that located in the far distance, the image contents nearby to the camera position become blur while the farther focus has the obvious texture cues. This correlation between blur cues and texture cues can be interpreted as the depth cues. The monocular depth cue is derived from an image which is characterized with the texture cues and the blur cues. The texture cues mean the density of the edge inside the block. The blur cues are the degree of blur and the sharpness of the edge. And in our research, we mainly focus on the local cues to make machine computation easier in one of the calculation loops. Starting from this point, we are targeting to enhance the feature extraction effectiveness for the texture and blur cues.

Consequently, the below relevant concepts are introduced. The multi-resolution of image representation as the coarse-to-fine analysis is shortly mentioned.

Principal Component Analysis (PCA) is not described in detail because it is introduced in chapter 2 as the popular method of extracting features from image data. PCA is the image data preparation phase in this study work. It extracts the edge lines through the eigenvalue and eigenvector analysis of image covariance matrix. DCT is applied to identify the feature distribution density of the edge lines for the object image. The combined application of PCA and DCT is carried out in this research work.

4.2.1 Coarse-to-fine analysis

An image can be resized with different resolutions and the resized images can be represented as a pyramid structure. In the local analysis, the better information coverage can be obtained by using the pyramid structure (Fig. 4-2) approach. The different resolution images are obtained by re-sizing the original image to smaller pixels. In this study, we define the 4 resolution levels from Level 0 to Level 3. Conceptually, the multi-resolution images provide the probability of feature extraction from the limited image block selection. The upper level of the pyramids, i.e., the low-resolution coarse level, is ideal to get an overview of the image scene. While the details can be found down the pyramid at higher resolution level.

This is namely the coarse-to-fine image analysis process, and this method is helpful to augment the feature extraction of the blur cues and texture cues.

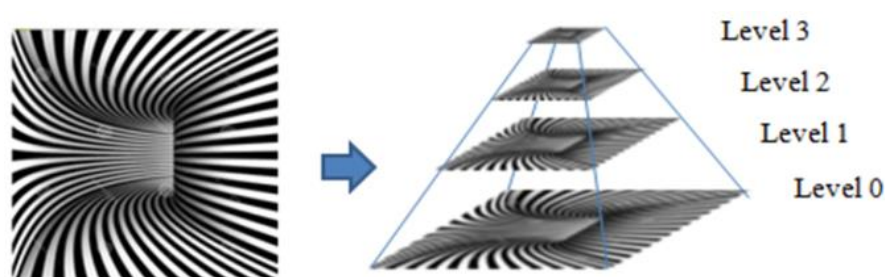


Figure 4-2 Image pyramid of coarse-to-fine

4.2.2 PCA for image processing

4.2.2.1 Principal component analysis

PCA is a statistical method, and it converts a set of variables that may be relevant by orthogonal transform into a set of linearly irrelevant variables. The converted group

of variables is called as the principal component. When large multivariate data sets are analyzed, it is often desirable to reduce their dimension. PCA is the appropriated technique for doing this. There are some choices associated with the technique which is covariance or correlation. PCA provides a method for data analysis and pattern recognition which is often used in signal and image processing. Because of its statistical technique, PCA is widely used in signal processing to the data dimension reduction or to the data feature extraction, such as edge line. PCA can also extract the orientation information of the edge lines. In the process of image feature extraction, PCA is often used to figure out the main eigen-characteristics for the image data sets through covariance matrix analysis. Suppose the points are on an image in x - y axis coordinate, from the point data, we can calculate the variance $C(x,x)$ in the x -direction and the variance $C(y,y)$ in the y -direction. However, the horizontal spread and the vertical spread of the data do not explain the clear diagonal correlation. If the x -value of a data point increases, the y -value increases also, resulting in a positive correlation. This correlation can be captured by extending the notion of variance to what is called as the “covariance” of the data: $C(x,x)$, $C(y,y)$, $C(x,y)$ and $C(y,x)$. These four values can be summarized in a matrix, which is called as the covariance matrix Eq(4-1):

$$C(x, y) = \begin{bmatrix} c(x, x) & c(x, y) \\ c(y, x) & c(y, y) \end{bmatrix} \quad (4-1)$$

The computation result of covariance matrix is a 2×2 matrix. From this covariance matrix, two Eigenvalues λ_1 and λ_2 , and two Eigenvectors e_1 and e_2 , can be obtained. Then the line characteristics can be derived from the Eigenvectors and Eigenvalues. The illustration of PCA analysis is shown in Fig. 4-3, where the eigenvalue and eigenvector can be simply listed.

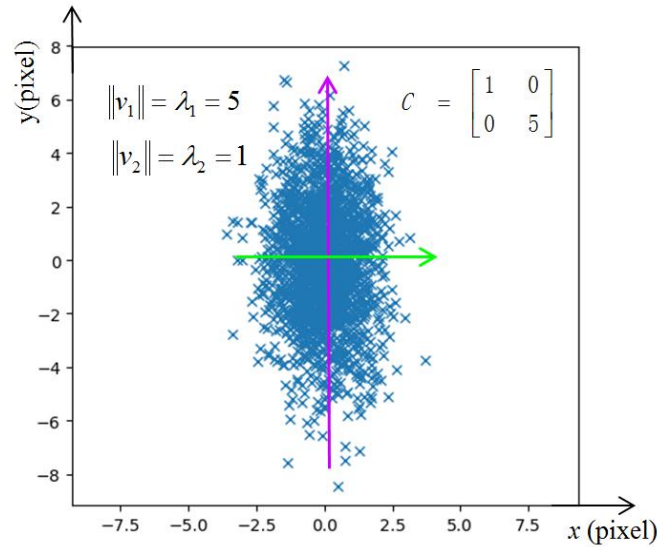


Figure 4-3 PCA analysis

The largest eigenvector of the covariance matrix always points into the direction of the largest variance of the data, and the magnitude of this vector is equal to the corresponding eigenvalue. The second largest eigenvector is always orthogonal to the largest eigenvector, and points into the direction of the second largest spread of the data.

In Fig. 4-4, the visual result of edge extraction is shown from the original image to PCA result.

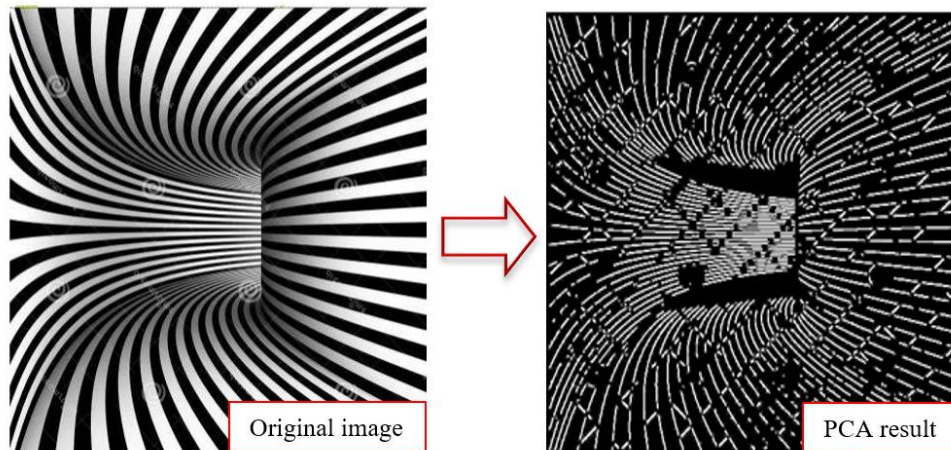


Figure 4-4 PCA visual result

4.2.2.2 Four category proposal

We proposed that all the edge lines extracted by PCA in the image can be separated into four categories (Fig. 4-5) through the computed line orientation according to eigenvector directions. As far as we know there is no such research available in which

the PCA analysis is applying the orientation information from covariance matrix eigenvector. Due to this processing, the density of the edge lines in spatial frequency domain can be intensively expressed in each category. Combined with the coarse-to-fine analysis approach, the obtained edge lines located in four categories by PCA are very helpful to extract the interested depth cue due to the enhanced effectiveness of spatial frequency analysis process. The orientation categories are described as the following. Here the angle information of an edge line segment is contained in the eigenvector.

Take eigenvector v_1 and eigenvalue λ_1 as the example:

$$v_1 = (x, y) = (\lambda_1 \cos \theta, \lambda_1 \sin \theta) \quad (4-2)$$

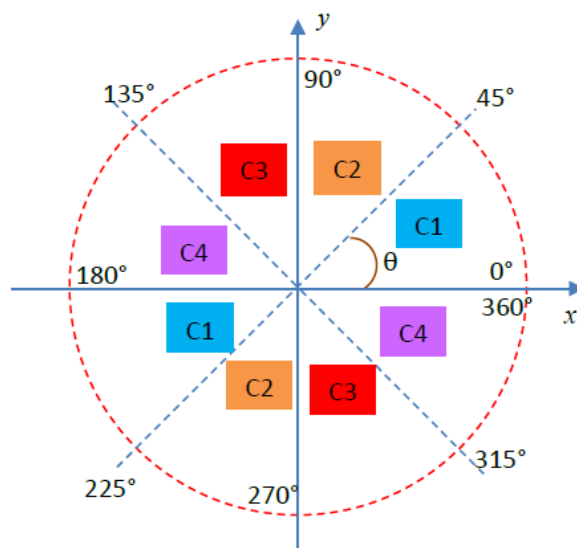


Figure 4-5 Category for edge lines

Accordingly, the edge lines could be defined as 4 categories. The correspondent results are demonstrated by later Fig. 4-13.

Category 1 (C1):

$$0^\circ \leq \theta < 45^\circ: \quad \sin \theta \in \left(0, \frac{\sqrt{2}}{2}\right) \quad \cos \theta \in \left(\frac{\sqrt{2}}{2}, 1\right) \quad (4-3)$$

$$180^\circ \leq \theta < 225^\circ: \quad \sin \theta \in \left(-\frac{\sqrt{2}}{2}, 0\right) \quad \cos \theta \in \left(-1, -\frac{\sqrt{2}}{2}\right) \quad (4-4)$$

Category 2 (C2):

$$45^\circ \leq \theta < 90^\circ: \quad \sin \theta \in \left(\frac{\sqrt{2}}{2}, 1 \right) \quad \cos \theta \in \left(0, \frac{\sqrt{2}}{2} \right) \quad (4-5)$$

$$225^\circ \leq \theta < 270^\circ: \quad \sin \theta \in \left(-1, -\frac{\sqrt{2}}{2} \right) \quad \cos \theta \in \left(-\frac{\sqrt{2}}{2}, 0 \right) \quad (4-6)$$

Category 3 (C3):

$$90^\circ \leq \theta < 135^\circ: \quad \sin \theta \in \left(\frac{\sqrt{2}}{2}, 1 \right) \quad \cos \theta \in \left(-\frac{\sqrt{2}}{2}, 0 \right) \quad (4-7)$$

$$270^\circ \leq \theta < 315^\circ: \quad \sin \theta \in \left(-1, -\frac{\sqrt{2}}{2} \right) \quad \cos \theta \in \left(0, \frac{\sqrt{2}}{2} \right) \quad (4-8)$$

Category 4 (C4):

$$135^\circ \leq \theta < 180^\circ: \quad \sin \theta \in \left(0, \frac{\sqrt{2}}{2} \right) \quad \cos \theta \in \left(-1, -\frac{\sqrt{2}}{2} \right) \quad (4-9)$$

$$315^\circ \leq \theta < 360^\circ: \quad \sin \theta \in \left(-\frac{\sqrt{2}}{2}, 0 \right) \quad \cos \theta \in \left(\frac{\sqrt{2}}{2}, 1 \right) \quad (4-10)$$

4.2.3 Edge Line Density Extraction by Discrete Cosine Transform

4.2.3.1 Discrete Cosine Transform

To get the edge line density and sharpness in frequency domain, the transformation is carried out to deal with the image in spatial domain. The discrete cosine transform helps separate the image into parts of differing importance with respect to the image's visual quality. The DCT is similar to the discrete Fourier transform: it transforms an image from the spatial domain to the frequency domain, but it can approximate lines well with fewer coefficients by using only real numbers. DCTs operate on real data with even symmetry.

The general equation for one dimensional (N data items) DCT is defined by the following Eq (4-11):

$$F(u) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \Lambda(i) \cdot \cos \left[\frac{\pi u}{2N} (2i+1) \right] f(i) \quad (4-11)$$

Where $\Lambda(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } i = 0 \\ 1 & \text{otherwise} \end{cases}$. The values $F(u)$ are called as DCT coefficients

of Λ .

The general equation for a 2-D (N by M image) DCT is defined by the following equation Eq (4-12):

$$F(u, v) = \sqrt{\frac{2}{N}} \sqrt{\frac{2}{M}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i) \Lambda(j) \cdot \cos\left[\frac{\pi u}{2N}(2i+1)\right] \cdot \cos\left[\frac{\pi v}{2M}(2j+1)\right] f(i, j) \quad (4-12)$$

Where $\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$ (ξ stands for either i or j). The values $F(u, v)$

are called DCT coefficients of Λ . In this study, the spatial frequency response is calculated by DCT.

4.2.3.2 Line density expressed by DCT spectrum

With the above theory introduction, the following consequent studies are conducted. As the binarized image texture resulted from PCA process is calculated by DCT, the edge line density in the image can be expressed by the spectrum density of spatial frequency. In other words, the DCT is to transform the edge distribution of the image into the spatial frequency distribution.

DCT is used to calculate the spatial frequency of the selected image block. According to the property of DCT, the low frequency is concentrated on the left top of the spectrum, and the high frequency is concentrated on the right bottom of the spectrum.

The low frequency indicates the sharpness of the edges, which is used to evaluate the blur cues. Both low and high frequency is used to evaluate the density of the edge inside the block. As an example, the spatial frequency illustration in DCT spectrum is shown in Fig. 4-6.

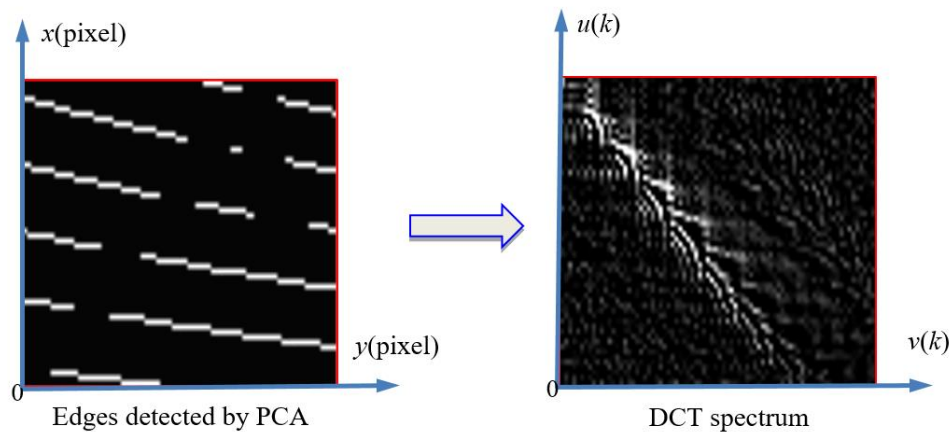


Figure 4-6 DCT spectrum for edge detection result

Here k means the wave number period in unit length, v means the horizontal frequency of 2-D waves, and u means the vertical frequency of 2-D wave.

To process the local cues obtained from coarse-to-fine representation, we apply a sliding window to compute the spatial frequency response. In coarse-to-fine representation, a window is defined appropriately to separate each resolution level into integer number. Then DCT is computed from one window to next window shown in Fig. 4-7. And the explicit example of DCT result is shown in Fig. 4-8. CNN takes the DCT results as its input and the depth inference comes out from the deep learning process.

In the study, DCT has more obvious feature. In the DCT results, the low frequency density part concentrates on the left top of the DCT map, and high frequency density part concentrates on the right bottom of the DCT map. This frequency distribution feature is fed to CNN for deep learning.

The image data processing algorithm PCA and spatial frequency analysis method DCT are already described in the above; CNN, which is used to estimate the depth cues, is consequently and briefly described in the following session.

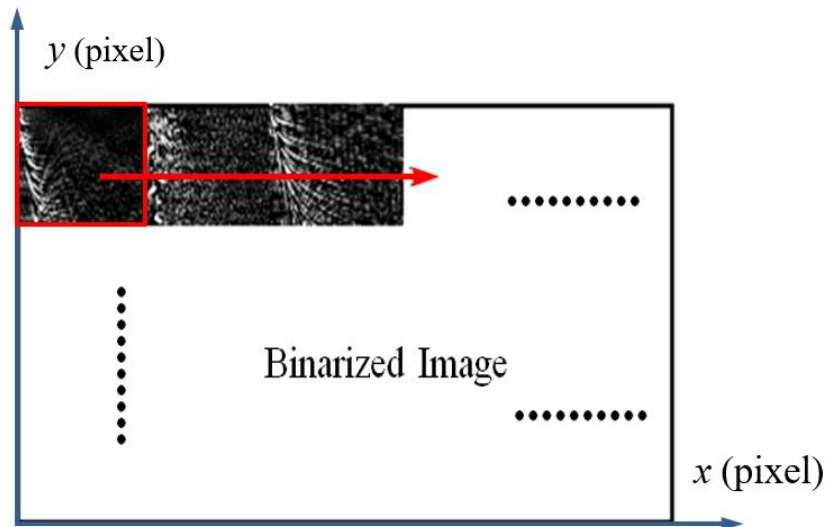


Figure 4-7 Spatial frequency calculation procedure

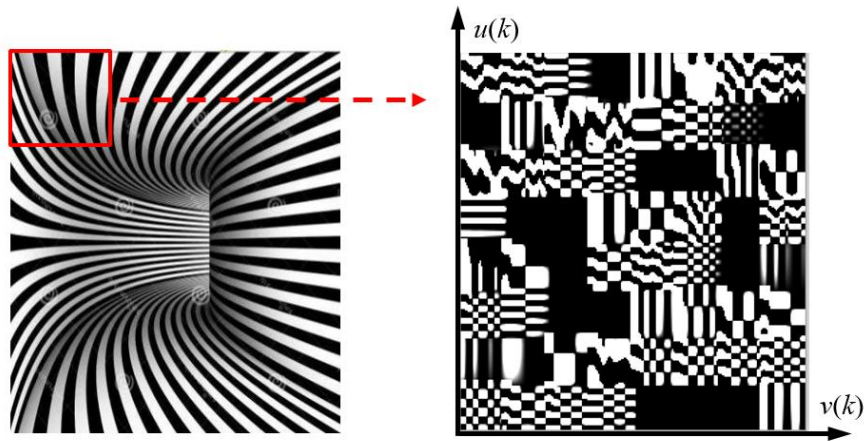


Figure 4-8 Block selection and its DCT result

4.3 Depth cue derived by CNN

CNN is one of the most powerful deep learning neural networks. Especially in the image processing application, CNN performance is highly evaluated. Through a lot of structure and performance improvements, CNN has become a widely used method. The advantages of CNN in image processing are the followings:

- (1) can reduce the number of weights in the calculation process.
- (2) can be robust in the object recognition for a slight distortion, deformation, and other interference.
- (3) has the characteristics of automatic learning and feature induction.
- (4) is not sensitive to object position change in the image.

4.3.1 Convolutional Neural Network

There are mainly three Neural Layers in CNN structure as introduced in previous session:

- (1) Convolutional Layer.
- (2) Pooling Layer.
- (3) Fully-Connected Layer.

The general connection layer is usually used in the multi-layer sensing neural network. The main role is based on the previous convolution layer and the pooling layer to extract the feature vector to deal with the image classification. As shown in Fig. 4-9, after convolutional layers and max pooling layers, the fully-connected layers come up. The weighting and bias process is operated in the fully-connected layer.

The outline of a convolutional neural network structure is shown in Fig. 4-9.

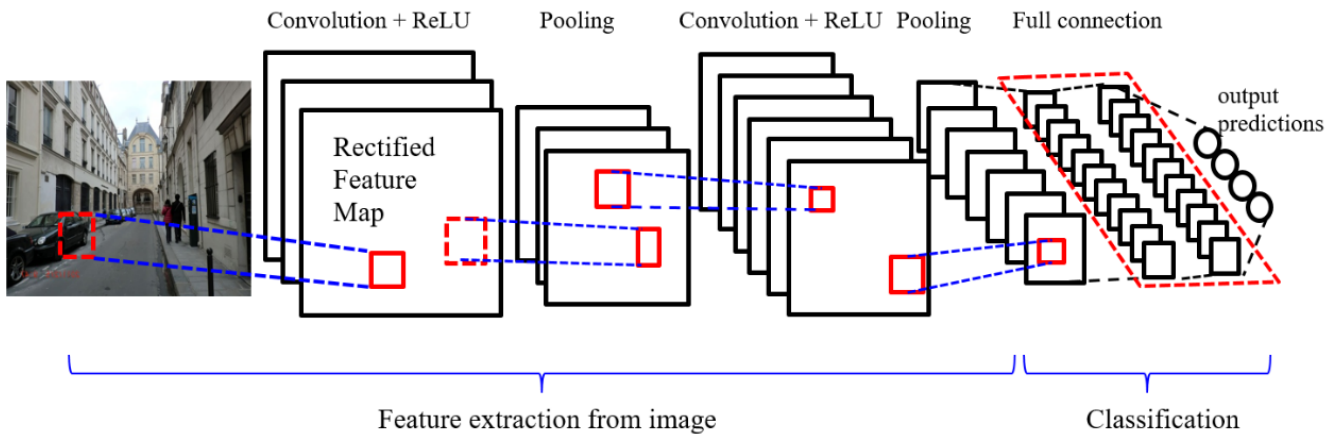


Figure 4-9 CNN structure

4.3.2 Proposed Structure Framework

Fig. 4-10 is the proposed structure framework. In the framework, texture cue path and blur cue path are the pre-processing phase to extract texture feature and blur feature respectively.

CNN is trained to learn the far and near information, which is the basis to classify the depth information of “near” or “far”. The depth estimation is presented as the comparison result of two different blocks on the image.

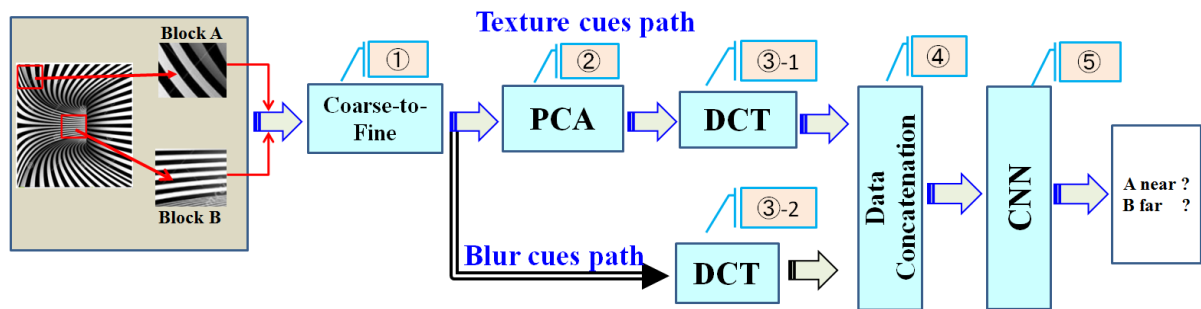


Figure 4-10 Proposed structure framework

This structure framework flow is described as the below:

Two blocks on 2-D original image are selected as the input. The block selection is arranged by shifting windows. These two blocks are marked as Block A and Block B. In real application, one of the blocks will be fixed in the image, so the distance information can be estimated.

- ① Apply the coarse-to-fine analysis to the blocks which are selected from the image and get different resolution pictures. In this study, 4 level resolutions are the definition

for the coarse-to-fine representation process. Then there are 4 resolution pictures consequently in each block.

② In the texture cue path, PCA with 4 category process is applied to Block A and Block B. And the PCA results are calculated by DCT (③-1) to get the feature map.

③-2 In the blur cue path, PCA is not applied to keep the blur features of Block A and Block B. DCT calculation is directly conducted to get the feature map.

④ Concatenate these DCT feature maps. And these DCT maps are aggregated together.

⑤ Through concatenation layer, the output is obtained through CNN deep learning process.

The detail break down of this mainstream flow chart is described in the following session.

4.3.2.1 Coarse-to-fine analysis

The image is pre-processed by the coarse-to-fine analysis with a multi-resolution image representation, and 4 resolution levels of the image are defined in this study. Coarse-to-fine representation can cover more information for image processing.

In this study, the Block A and Block B are selected from an original image. Then apply the coarse-to-fine analysis to get 4 level resolutions (referring to Fig. 4-11). This method is applied to get more texture cue and blur cue information.

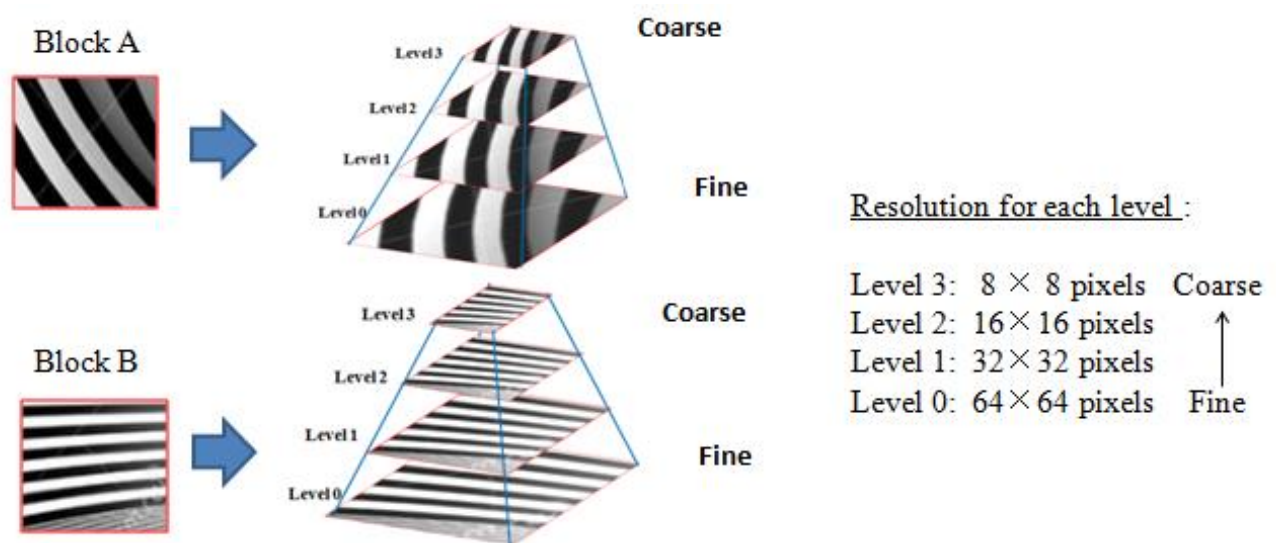


Figure 4-11 Coarse-to-fine processing

4.3.2.2 Texture cue path

PCA is used to extract the edge lines from image. The 4 category edges computed by PCA are used to classify the edge line directions to 4 segments as Category 1, Category 2, Category 3, and Category 4. The 4 category PCA results are shown by Fig. 4-13.

Image can be analyzed through different categories in order to have more information in certain direction.

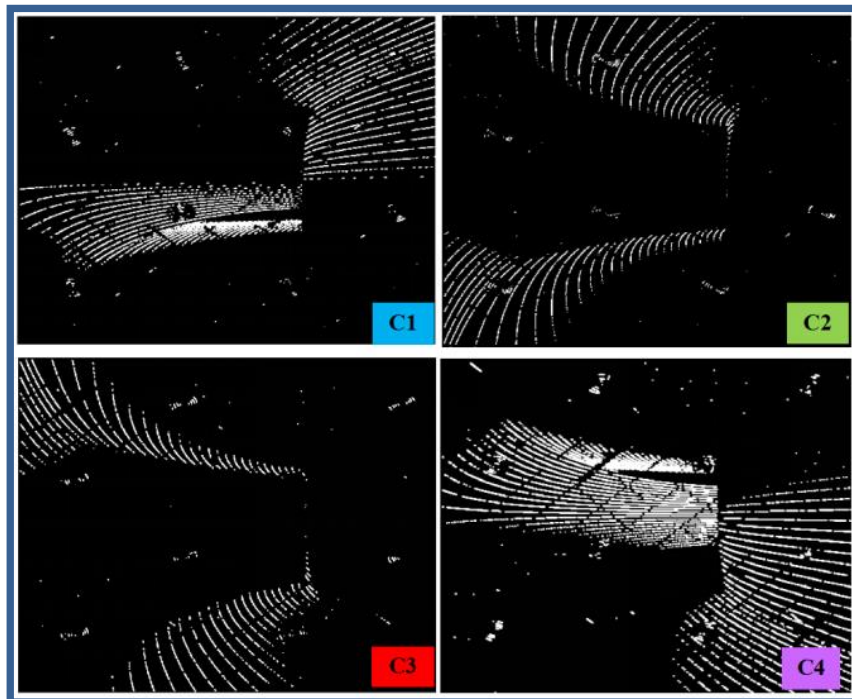


Figure 4-12 PCA result by four categories

After the PCA pre-process, DCT is used to calculate the spatial frequency of the selected block. The low frequency is used to evaluate degree of blur cues, i.e., the degree of blur and the sharpness of the edge. Both the low and the high frequency are used to evaluate the density of the edge inside the block, which is representing the texture cue.

The below is the visualized demonstration for DCT results of a block at each resolution level (Fig. 4-13).

As we are processing the coarse-to-fine analysis, in Fig. 4-14, the block A partially demonstrates the visual understanding for the steps of PCA to DCT representations

regarding to category 1. Similarly, category 2, 3, 4 are processed with the same approach.

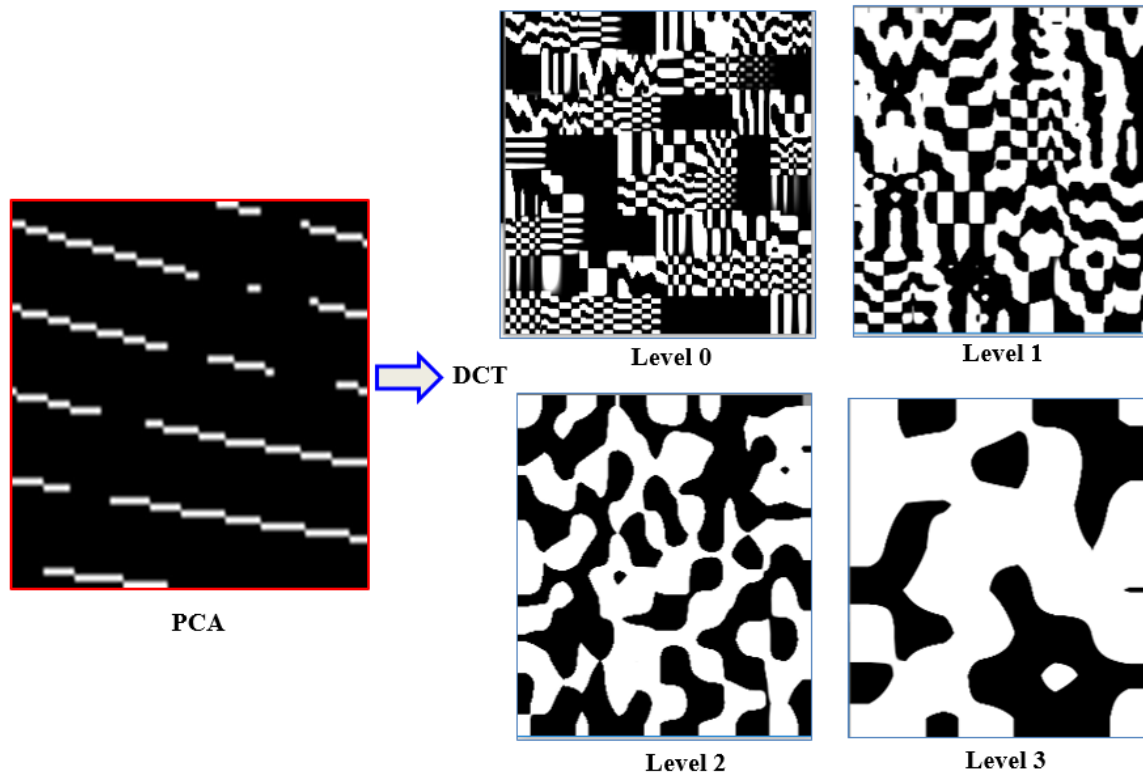


Figure 4-13 DCT results: a block at each resolution level

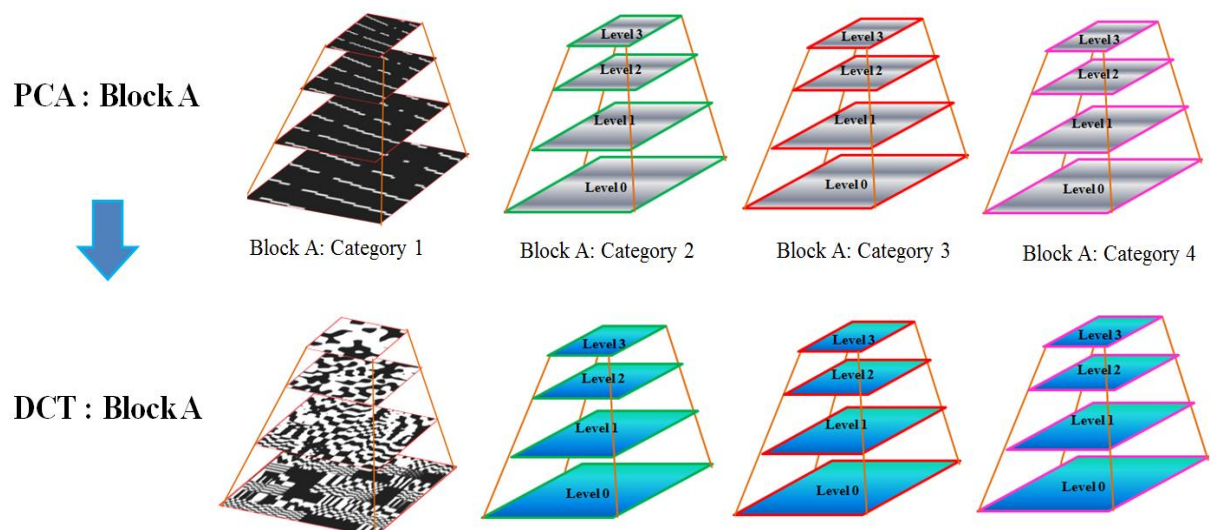


Figure 4-14 PCA-DCT: 4 resolution representations of 4 categories

4.3.2.3 Blur cue path

Referring to Fig. 4-10, in the blur cue path, PCA is not applied to keep the blur features of Block A and Block B. DCT is directly applied to the original image. The same approach as Fig. 4-15 is conducted to calculate DCT for different resolutions of Block A. Similarly same approach is conducted to apply DCT for different resolutions of Block B.

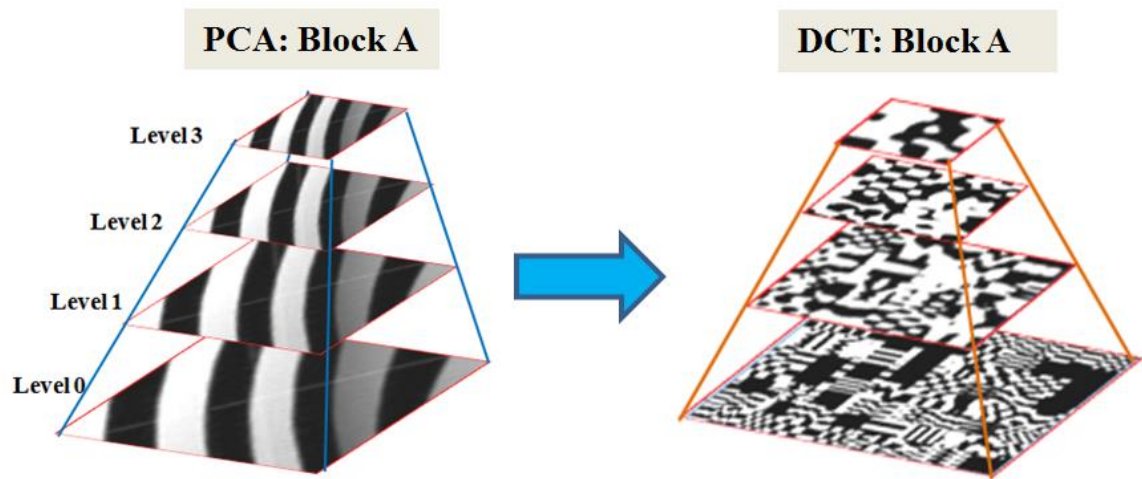


Figure 4-15 Blur cue path DCT

4.3.2.4 Data Concatenation

Referring to data concatenation in Fig. 4-16, the DCT results need to be concatenated for CNN input. The data concatenation is the data aggregation array: $C \times H \times W = 2 \times 64 \times 64$

Fig. 4-16(a) shows the data aggregation for Block A category 1 and Block B category 1 at each resolution; consequently, it is the same the process for category 2, 3 and 4 (the figure demonstration is omitted) at each resolution level. Similarly, DCTs of block A and block B at each resolution level are concatenated as shown in Fig. 4-16(b).

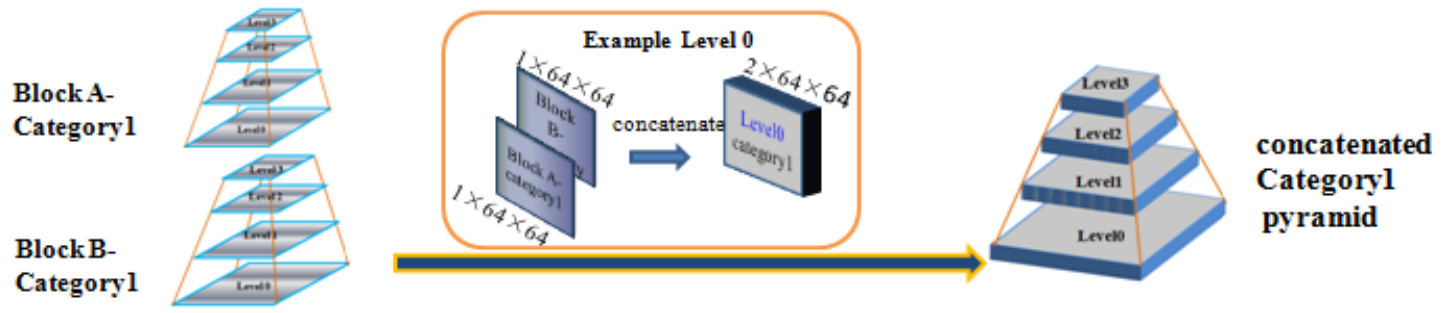


Figure 4-16(a) Concatenation

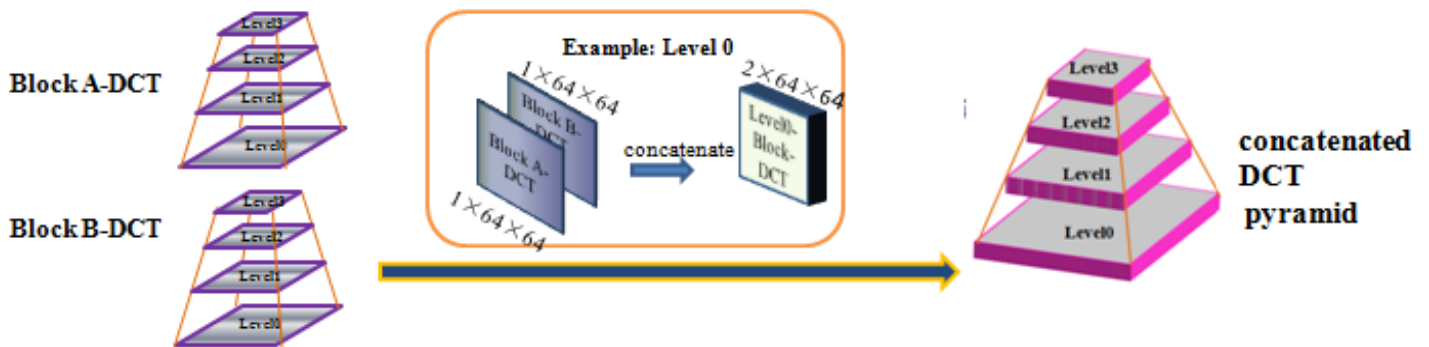


Figure 4-16(b) Concatenation

Fig. 4-17 shows the detail concatenation example of $P \times P$ pixels block size and 8×8 pixels kernel size.

$$C_{kl} = \sum_{n=1}^4 \sum_{i=1}^8 \sum_{j=1}^8 m_{ij}^{(n)} x_{ij}^{(n)} \quad (4-13)$$

Here $1 \leq k \leq P-8+1$, $1 \leq l \leq P-8+1$;

$m_{ij}^{(n)}$: the element of the filter mask at coordinate i and j in the block; n is the order number of the matrix.

C_{kl} : convolution result at position k, l .

$x_{ij}^{(n)}$: element of the filter at coordinate $k-i$ and $l-j$

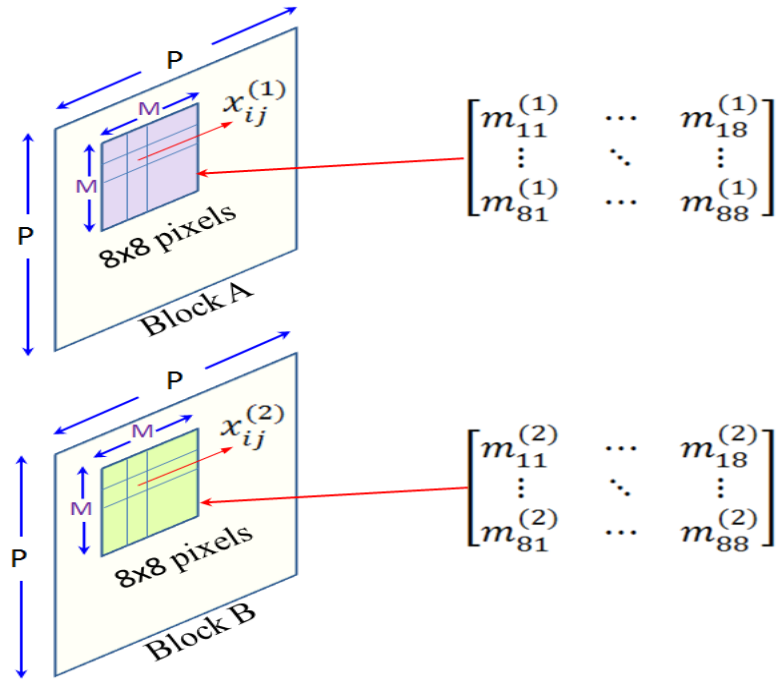


Figure 4-17 Concatenation illustrations

The visual representation for the convolution is shown in Fig. 4-18.

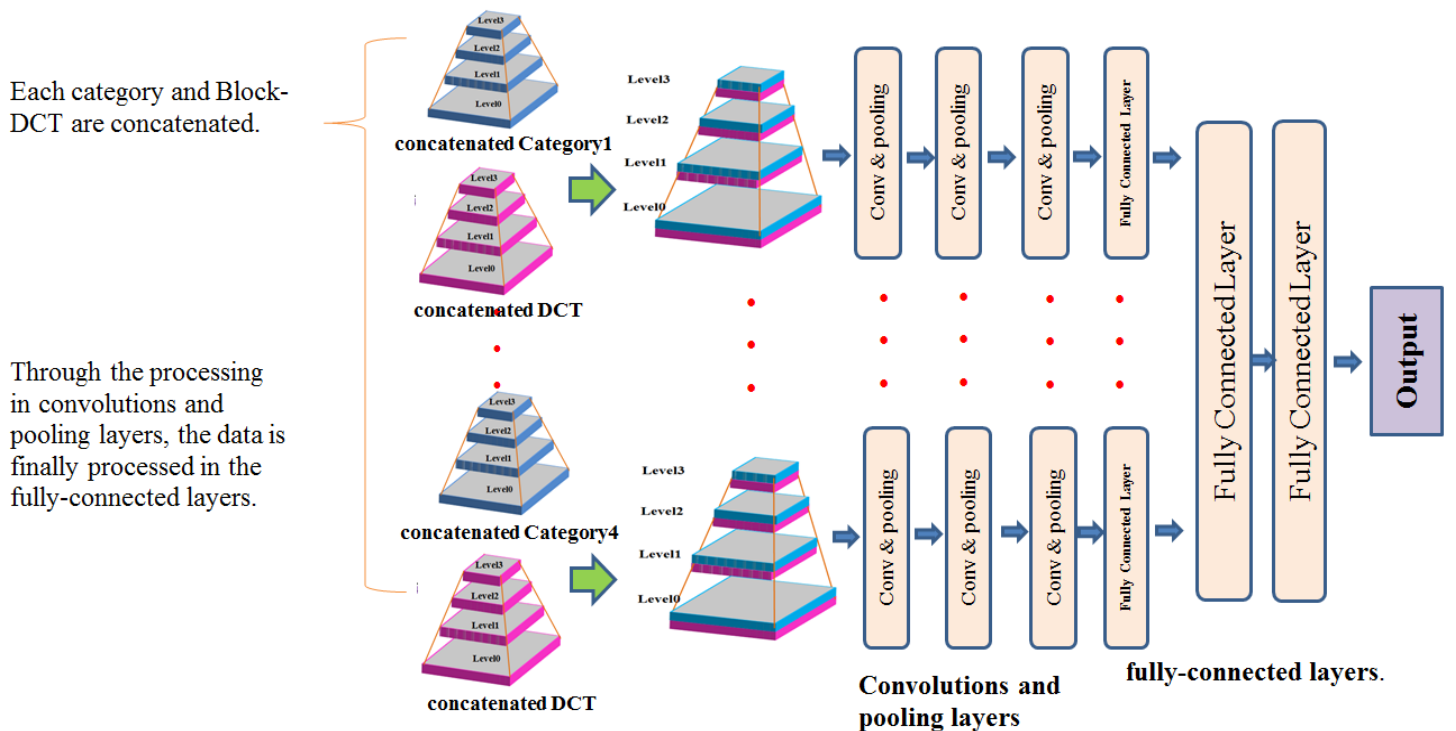


Figure 4-18 Diagram frame of convolution

In order to have the overview for the entire calculation architecture, the diagram frame for data pre-processing of PCA-DCT, data concatenation and CNN frame diagram are shown in Fig. 4-19. In the pre-processing frame, the texture cue path and

blur cue path are expressed. And then the data concatenation is aggregated for CNN input. The detail for CNN calculation configuration is shown in Fig. 4-20.

4.3.3 CNN Diagram and Parameters

The target is to testify the proposed approach of using texture and blur as the cues to estimate the depth information. So, the training image data needs blur and texture on the image and the proper image data of monocular camera is selected.

This net is setup in the Caffe environment, and the details are introduced by the example of category 1, at resolution level 0:

- 1) Image data is $4 \times 64 \times 64$ which is coming from the concatenation result.
- 2) Convolution layer: convolves the input image with a set of filters, and each filter produces one feature map in the output image. In this CNN structure, 3 layers are used.
- 3) ReLU is applied to get a better Stochastic Gradient Descent (SGD) convergent speed and to prevent from going to dead zone without convergence.
- 4) Pooling layer: reduces the output feature vector of convolution layer by sub-sampling.
- 5) Inner Product: the fully-connected layer.

In the consequent section, the CNN model and the experiment results are provided to verify the effective parameter scenario.

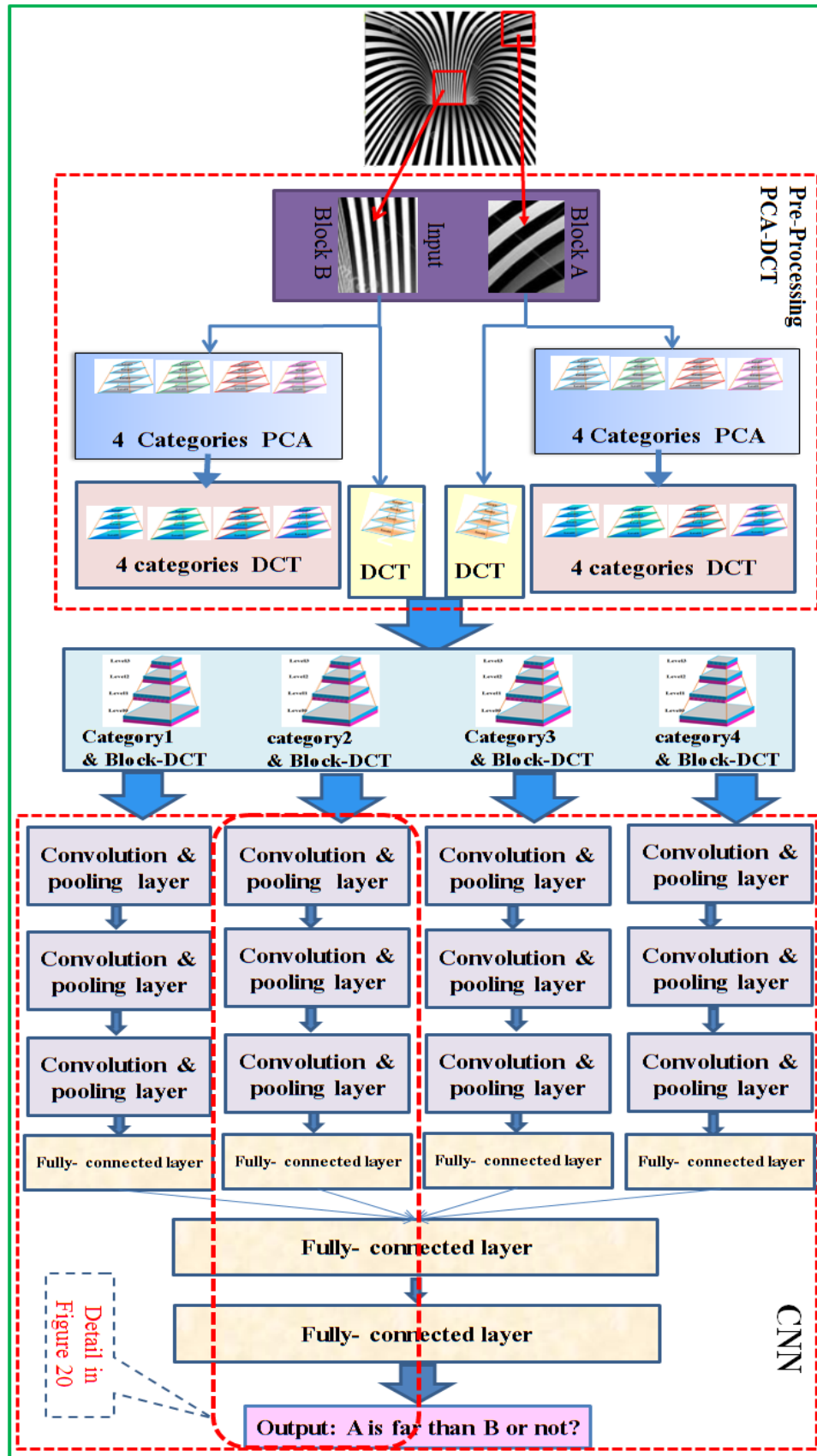


Figure 4-19 Calculation architecture

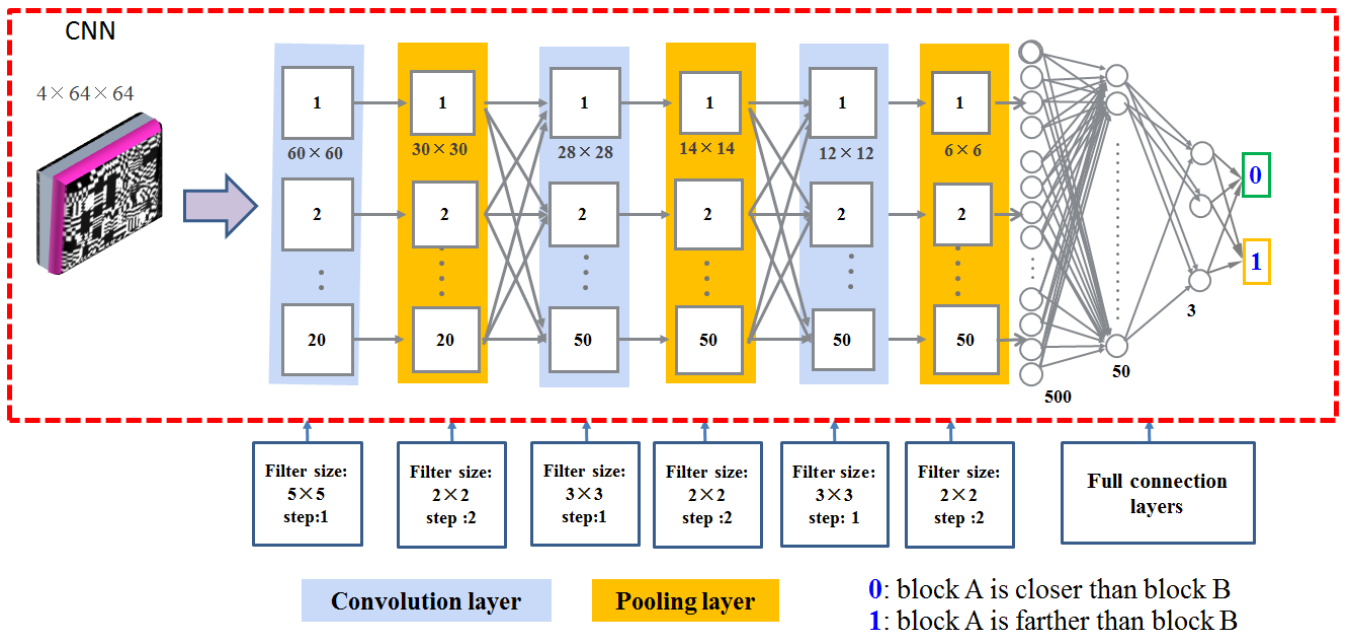


Figure 4-20 CNN diagram frame in detail procedure

4.4 Evaluation of the proposed method

The investigations of the following test scenarios are carried out to evaluate the effectiveness of the proposed method.

4.4.1 Experimental setup and testing

The procedure of the experiment setting is listed as the below.

- (1) Train the proposed CNN
- (2) Choose blocks on original images for image dataset
- (3) Test different images

For the purpose of CNN model training, the following pictures are selected.

Data preparation: Mesh all of training pictures (800×600 pixels) as 10×10 block unit (Fig. 4-21(a)) and label the distance of each mesh segment. The split segments are shown in Fig. 4-21(b).

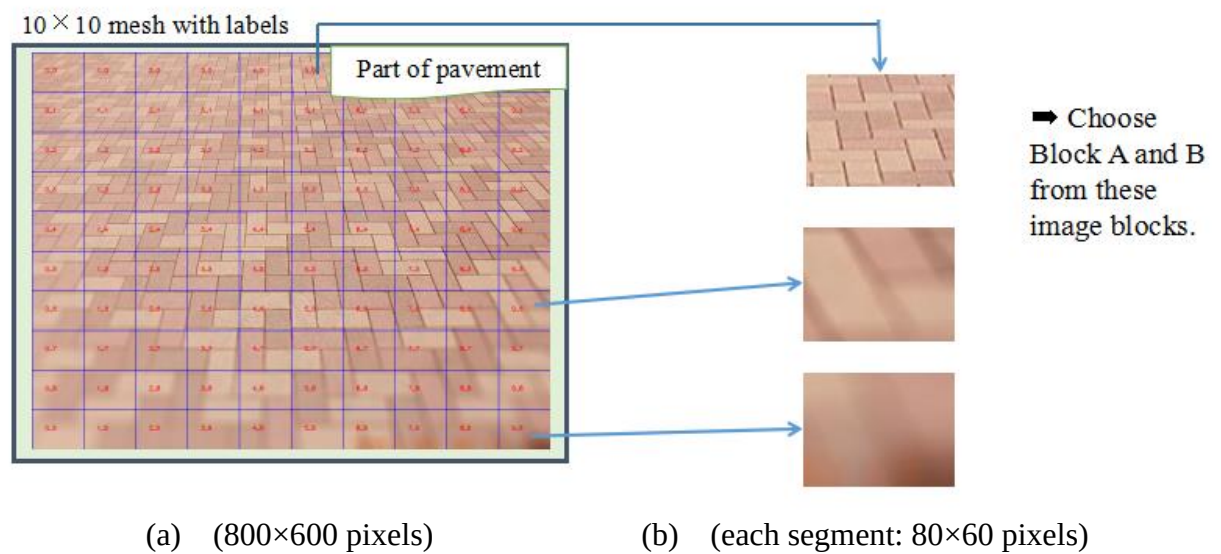


Fig 4-21 10×10 Mesh and Segment

Each block has its own distance label. Two different blocks inside one picture is taken as one pair which is selected by program label index. Therefore, each picture has total $100 \times 99 = 9900$ pairs. In each test, 500 pairs block among the 9900 pairs are used for model training calculation.

4.4.2 Testing for the proposed method

The details of the test scenarios can be referred to the reference [13] in which the effectiveness of the proposed method is demonstrated. Here just shows the test image

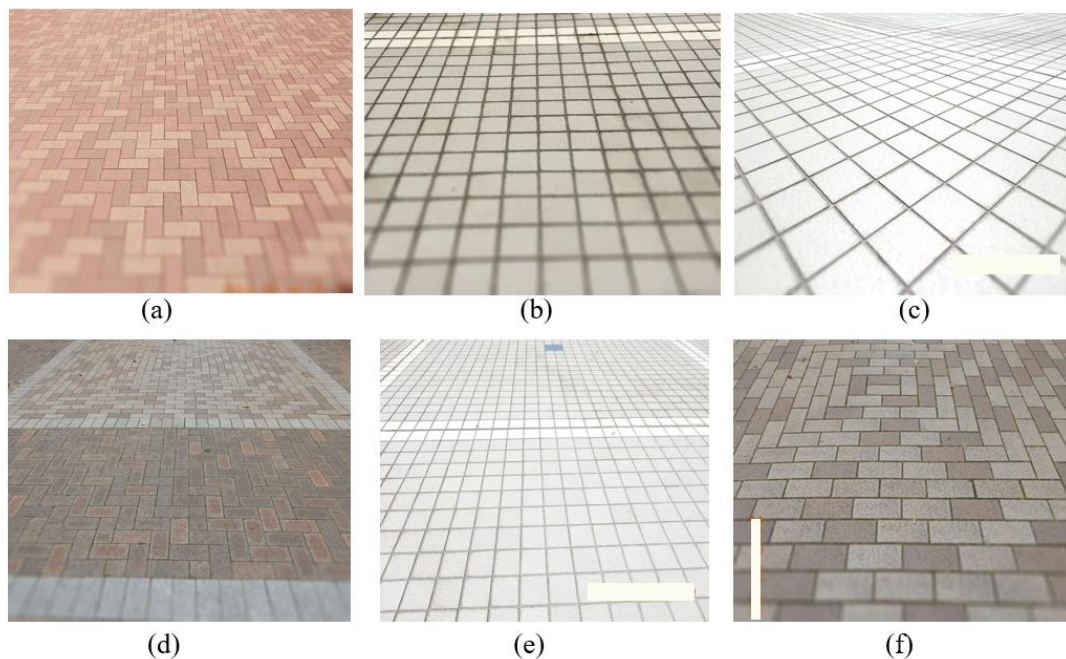


Figure 4-22 Image data with clear texture

data as Fig. 4-22 and the experiment results in Table 4-1. The average accuracy 94.96% verifies that the proposed approach is effective with regarding to the images relatively close to the theoretic assumption.

Table 4-1 Testing result of the sample image

Images for the test	(a)	(b)	(c)	(d)	(e)	(f)	average
Accuracy of the test	93.13%	95.25%	93.63%	92.38%	97.75%	97.63%	94.96%

4.4.3 Practical application study for vehicle blind spot detection

As per the focus point for the monocular camera is assumed to be located at the far beyond vehicle side mirror view, the images taken by the rear side view camera have these features at the driving condition: near area is blur zone and the far from blind spot area is clear zone.

These features are compatible to the proposed algorithm approach. In this case, the experiment is set to ensure that the proposed method is applicable to the vehicle blind spot detection and the effectiveness is verified by the road traffic images.

4.4.3.1 Experimental setup

The camera measurement setup is shown in Fig. 4-23. The camera is equipped at the rear side view mirror, where it is compatible to the reality.



Figure 4-23 Measurement setup

Our proposed method is to detect the obstacles getting near to the ego vehicle in its blind spot area. The detected depth information is relative to the ego vehicle base block, which is pre-selected, referring to Fig. 4-24. All other blocks are calculated respectively by the sliding window shifting (refer to Fig. 4-7). And one of the possible nearest blocks with higher accuracy comes out through the deep learning CNN (refer

to Fig. 4-19 and 4-20). Currently, the proposed method is not an algorithm to estimate the absolute distance value.

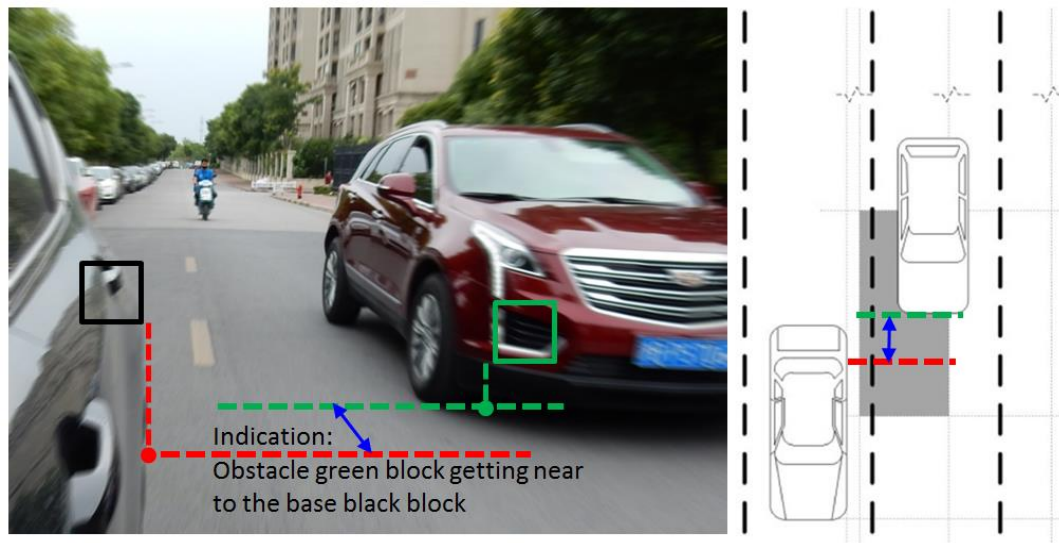


Figure 4-24 Depth information estimation

4.4.3.2 Testing of real traffic image data

As the focus point of the monocular camera is assumed to be located at the far beyond vehicle side mirror view, the images taken by the rear side view camera have these features at the driving condition: near area is blur zone and the far from blind spot area is clear zone. These features are compatible to the proposed algorithm approach. 50 sample data (Fig. 4-25) is used to make the supervised CNN model training under the off-line condition. In this case, the training experiment is set to ensure that the proposed method is applicable to the vehicle blind spot detection and the effectiveness is verified by the road traffic test image data.

As the results, the training model generates the acceptable results reaching to the average accuracy 95.3% (Table 4-1).

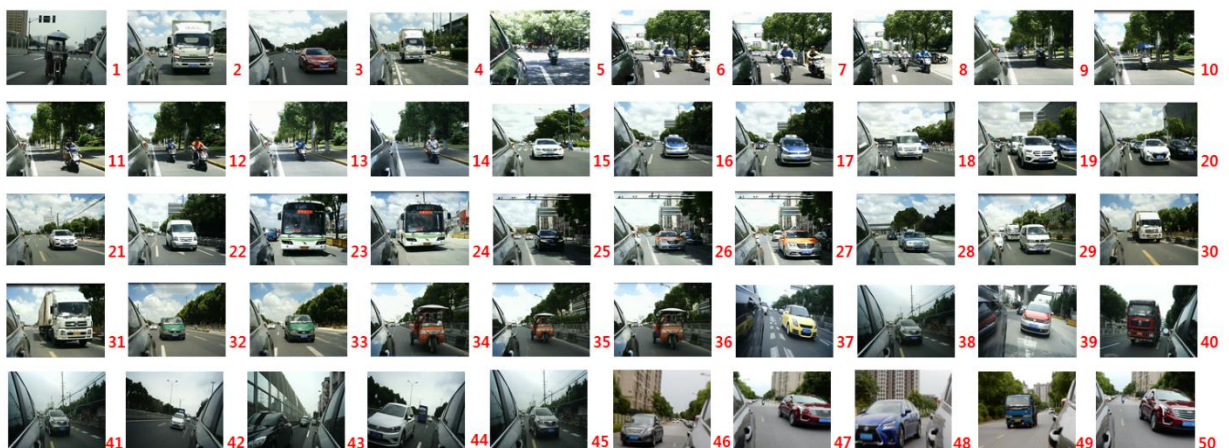


Figure 4-25 sample images for CNN model training

Table 4-1 Training result accuracy

Table 1: Training result accuracy											
	#1								#10		
#1	93.7	96.5	95.8	96.3	94.6	97.8	96.2	94.8	95.5	94.2	#10
#11	95.6	94.8	95.3	94.8	96.7	94.4	95.6	97.8	96.7	95.9	#20
#21	96.8	97.5	96.5	95.7	96.1	95.3	96.1	92.6	97.2	95.8	#30
#31	96.4	97.2	96.3	94.6	91.9	93.7	95.2	94.4	93.1	95.3	#40
#41	94.8	94.3	97	95.4	93.6	95.5	94.8	93.7	96.4	95.1	#50

With the trained model, the test scenarios are verified by 20 BSD images (Fig. 4-26). The test results are shown in Table 4-2. This average accuracy 87.4% verifies that the proposed algorithm is effective with regarding to the images relatively close to the theoretic assumption.



Figure 4-26 Testing image data of BSD

Table 4-2 testing result accuracy

#1	88.3	87.1	87.3	87.6	85.2	89	86.2	86.4	85.7	89.6	#10
#11	87.5	86.4	87.1	88.2	85.6	89.9	87.2	89.1	87.4	86.9	#20

As the summary, in the relatively ideal test case, the CNN deep learning model can achieve the higher accuracy. And the use case for vehicle blind area detection is also effective with accuracy 87.4%, though it is understood that the performance needs to be improved to handle more complex vehicle environment. There is no doubt that much huge real road traffic data are required to verify and release the BSD function.

4.5 Summary and research in next step

4.5.1 Summary

The proposed method of obtaining the monocular depth cues is testified and its effectiveness is verified by the testing results. The deep learning CNN model is good enough for the ideal case images with satisfied accuracy. Using the same CNN model to adapt to the depth information classification for vehicle blind spot area detection, the effective result is obtained with the better accuracy. We could believe that the proposed method is potential for more complex driving situation. Some advantages and disadvantages are summarized as the below:

- (1) The proposed approach of the combination with coarse-to-fine and four category process enhanced the effectiveness of depth cue extraction.
- (2) The proposed method is robust as the detection is not limited to vehicle shape profiles. Any obstacles approaching near to the host vehicle can be detected because the method calculates the local cues to derive the relative depth information.
- (3) The proposed method is not related to the image color. But the detection performance is not good enough for the image taken in the condition of dark night; this is the common issue in image processing area.
- (4) Because of the shifting window block in a single image calculation for local cues, the total calculation cost is still the concerning due to the costly sliding window calculations and the current micro-processor capacity versus product cost.

4.5.2 The research in next step

More parameter sets and training data sets need to be applied to the proposed method under vehicle driving condition related to the performance of blind spot area detection. By using the huge real driving scene image data, the robust CNN model can be obtained through the deep learning effort. And then the more fidelity result with higher accuracy can be achieved. This outcome can be used as the warning trigger signal to the driver when he is operating lane change, while a car is in his blind spot area at neighbor driving lane. Due to the driving situation, on-line in real time blind spot detection is required and the algorithm calculation cost and efficiency still are quite challengeable. We are looking forward to the further optimized approach through the consequent research effort.

In addition to the above problems, in the process of exploring the depth information, we found that the depth information of the object actually endows the object with the characteristics of the scale variance. That is to say, due to the depth and distance information of the object, the object appears in different scales. Therefore, it is necessary to have scale-invariant features to realize object recognition. According to the above research work, the goal of the next research work is to develop a new algorithm different from the image pyramid method to extract the features of scale invariance of the object. By fusing the feature of scale invariance to deep CNN structure, the detection for the same object at different scales can be implemented. When using PCA to obtain texture information in previous work, we found that PCA can describe

the shape of the object, so we hope to obtain scale-invariant object shape information by exploring different methods of using PCA. The next chapter will discuss the recognition method of scale variant object.

Chapter 5. Object detection in different scales by using PCA and multi-pooling layers

5.1 Introduction of object detection

Object detection has been an active and challengeable topic in computer vision research area. Object detection supports a wide range of applications, and it can be classified as the detection of specific instances and the detection of broad categories. The first type aims to detect instances of a particular object, essentially subject to a matching problem. The second type is to detect instances of some predefined object categories. So far, much of the effort in the field of object detection has been focused on the detection of a single category, typically human faces, and pedestrians^[10], or a few specific categories to meet the real application needs.

Principle component analysis (PCA)^[17,18] is widely used for dimension reduction and feature extraction, such as edge lines. In the process of image feature extraction, PCA is often used to identify the main eigen-characteristics for the image dataset through covariance matrix analysis. The correlation matrix is of the importance with the implication of scale invariance to PCA, which is attempted to get rid of the problem of scale dependency by the derived features of eigenvalues and eigenvectors^[18]

5.2 Related research

In dealing with the problem of detecting objects at different scales, many approaches are tried to achieve the accurate results with less computation cost. In the literatures [13,14,19,20,21], pyramid structure is intensively used to extract features and it is applied to support object detection at different image scales. Image pyramid is usually obtained by down sampling or up sampling the original image multiple times. And then we can detect objects at different resolutions and get more comprehensive object detection results. Due to the dynamic real-world scene is keeping changing, especially the moving objects, images may be collected at different scales. Image pyramid has a wide range of applications in multi-scale object recognition and is useful to analyze and process images at different scales. Image pyramid is usually applied at the entrance of deep CNN^[13,14]. An image pyramid is a multi-scale representation of an image, where the original image is progressively down sampled to create a series of images at different resolutions. The pyramid typically starts with the original image at the highest resolution and progresses downwards to coarser resolutions. Reference studies [22,23]

apply multi-scale inputs sequentially from coarse-to-fine, while reference studies [24,25,26] directly resize the input for several scales and fuse the features from all the scales. The coarse-to-fine approach is a common strategy for using image pyramids in computer vision applications. The coarse levels of the pyramid provide a low-resolution overview of the image, which can be used to quickly identify large-scale features or regions of interest. The finer levels of the pyramid provide progressively more detailed information, which can be used to refine the analysis of the image and localize features with higher precision.

The reference study [19] proposed a recursive feature pyramid network (RFP), which use single scale features and is computed like featurized image pyramid. Feature pyramid is a basic component for detecting objects at different scales. By exploiting inherent multi-scale, pyramidal hierarchy of deep CNN can be used to construct feature pyramids with marginal extra cost. A top-down architecture with lateral connections is developed for building high-level semantic feature maps at all scales.

CNNs are typically designed to take in fixed-size inputs, which limits their ability to recognize objects at different scales^[27,28]. Due to the fixed input image size requirement, image data pre-conditioning becomes necessary.

Reference [71] is an important contribution to the field of computer vision, and it presents a technique called spatial pyramid pooling (SPP) that is proposed to improve the performance of deep CNNs in image recognition tasks. SPP is a pooling method that divides the input image into multiple levels and computes the pooling features for each level separately. The features from all the levels are then concatenated to form the final feature vector. This allows the network to recognize objects at different scales. SPP maintains spatial information by pooling in local sliding windows. The size of these sliding windows is proportional to the size of the image, so the number of windows is independent of the image size. The SPP strategy can be used to eliminate the requirement of fixed image size or image scale. That means SPP can generate a fixed-length representation regardless of image size or scale and pyramid pooling is also robust to object deformations. With these advantages, SPP proves the capability of accuracy improvement. The power of SPP is significant in object detection, as it computes the feature maps from the entire image only once, and then pool features in arbitrary regions to generate fixed-length representations for training the detectors. This method avoids repeatedly computing the convolutional features.

Principal component analysis (PCA)^[17,18] is a widely used technique in image processing and computer vision for feature extraction and dimensionality reduction. To detect scale variant objects, PCA can be applied at multiple scales, using different window sizes or image resolutions. This allows the algorithm to capture features at different levels of detail, and to detect objects of different sizes or scales.

As per the relevant reference paper survey, it is significant to improve the performance of object detection at different scales. Based on the theoretical analysis and practical study, we propose to use the normalized eigenvalues as the feature of scale invariance. The eigenvalues are calculated by PCA processing for the covariance matrix composed of convolutional feature output. Taking the deep CNN network VGG-16^[40] as the base structure, we replace spatial pyramid pooling layer with our modified multi-pooling layer which are located before fully connected layers; meanwhile, PCA process loop is connected parallel to the multi-pooling layer before fully connected layers. This modified network structure is expected to enhance the robustness for the same object detection at different scales. The proposed network will be described in next session. The remaining of this chapter is as the followings.

Session 5.3 describes the algorithm structure of the proposed deep CNN. And the derivation of normalized eigenvalue as the feature of scale invariance is demonstrated. In session 5.4, the network model verification for scale variant object detection is explained. Three network comparisons are given, as the proposed model has better performance than original VGG-16 model and VGG-16 based SPP model. VGG-16 based SPP model has better performance than VGG-16 model. The training dataset is composed of the scene pictures from real road vehicle driving, because it is the easy way to include the different object scales in the captured images. As the comparison with the state-of-the-art techniques, four networks are chosen to make the testing. A short discussion is included in session 5.5. Some conclusion remarks are included in the session 5.6.

5.3 Proposed algorithm structure

5.3.1 Brief of the CNN outline

What we proposed is to use a parallel structure of multi-pooling & PCA to extract the feature of scale invariance and then to recognize the object at different scales. To demonstrate the effectiveness of this parallel structure, referring to Fig. 5-1, we use VGG-16 as the base (Fig. 5-1①) neural network structure. The first five layers conv1~conv5 of the network consist of iterative convolution and pooling layers. The multi-pooling approach in Fig. 5-1③ is developed by modifying spatial pyramid pooling (SPP) network at pool_5 as shown in Fig. 5-1②. In parallel to multi-pooling layer, PCA

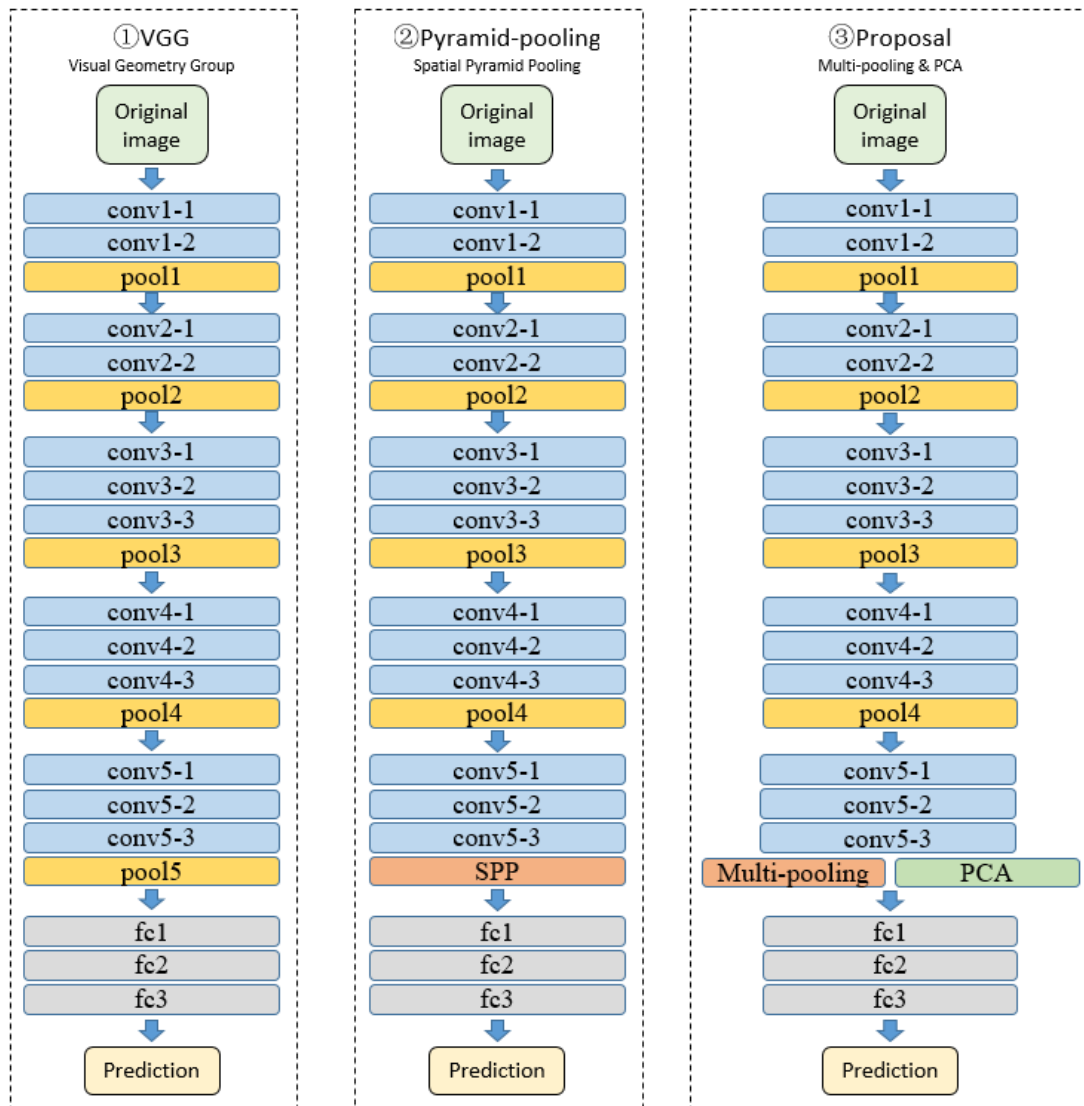


Figure 5-1 Network structures

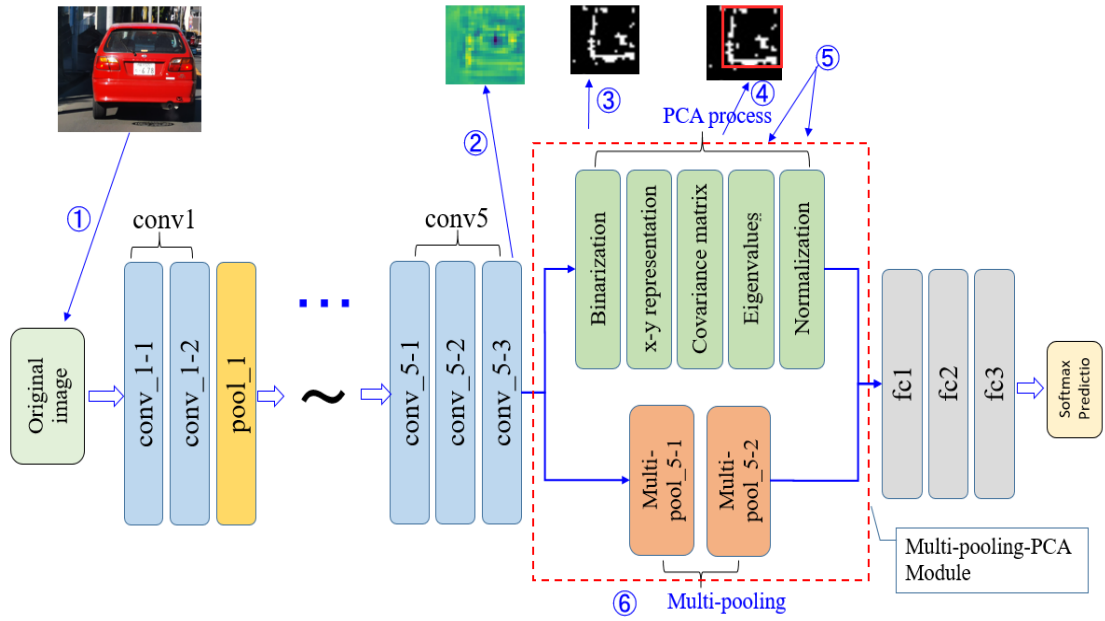


Figure 5-2 Demonstration on PCA process and feature of scale invariance process approach is located as shown in Fig. 5-1③. In Fig. 5-1③, at the last pooling layer, a stacked multi-pooling calculation is performed to achieve the same output formulation of fixed-length representation as that of SPP in Fig. 5-1②. We proposed the module (red dash line) of parallel routines of PCA process and multi-pooling layers to support the scale-invariant feature extraction, to augment the recognition robustness, referring to Fig. 5-2. This module provides the robust feature extraction of scale invariance. The PCA process performs covariance matrix analysis on the feature maps (Fig. 5-2 ②) resulted from the convolution layers and obtains the feature map eigenvalues and eigenvectors. In this study, we use the eigenvalues as scale-invariant reference information. The final three layers of this_proposed structure are fully connected layers. The network output is Softmax prediction of object recognition accuracy.

5.3.2 Description of multi-pooling layers

The visualized feature maps (Fig.5-2②) generated by convolution filters from conv1 to conv5 layer are shown in Fig. 5-2. These filters produce the strongest activation of car object images in the dataset. Some of the convolutional filters can be easily activated by some semantic contents, i.e., specific shapes, like circle or square, etc. Because the fully connected layers require a fixed length vector as input, so the fixed feature map size is necessary. To adopt deep networks for a regulated feature map

size, the last pooling layer “pool5” can be replaced with spatial pyramid pooling (Fig. 5-1 ②).

As shown in Fig. 5-3, in each sliding window, the response of each filter is aggregated together through max-pooling approach. In this approach, Step1 is a 2×2 max pooling, and Step 2 is a 4×4 max pooling. Spatial pyramid pooling parameter is 32.

With spatial pyramid pooling, the input image can be arbitrary sizes, allowing for arbitrary proportion and scales of the input. The input image can be any scale and apply the same deep network. When the input image is represented as different scales in SPP, the network with the same filter size will extract features at different scales. These different scales in SPP are important for the accuracy of deep networks. In these different scales, the coarsest pyramid layer has only one window, covering the entire

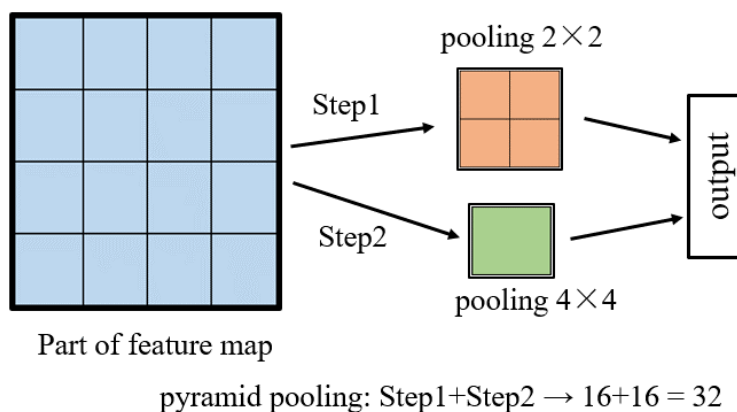
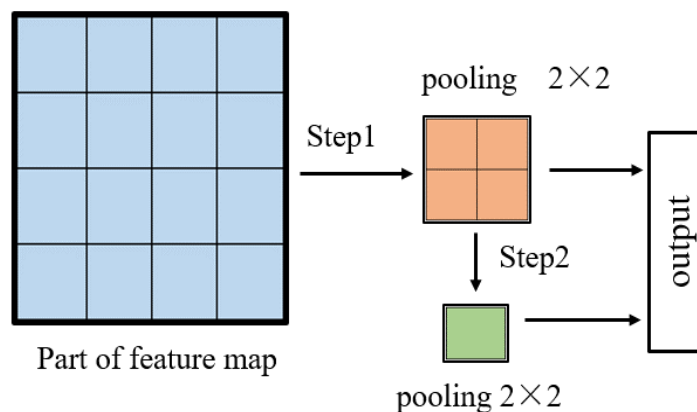


Figure 5-3 Pyramid pooling in SPP network



Multi-pooling parameters: Step1+Step2 $\rightarrow 16+4 = 20$

Figure 5-4 Multi-pooling in network architecture

image. This is actually a global pooling operation. As a result, the SPP includes both local and global features.

Corresponding to spatial pyramid pooling approach, to optimize spatial pyramid pooling, we propose to perform multiple pooling in a stacked computation manner to achieve the same output as that in Fig. 5-3, but with fewer parameters. The specific implementation is shown in Fig. 5-4.

In this multi-pooling approach, Step1 is a 2×2 max pooling, and Step 2 is a 2×2 max pooling of previous step. Multi-pooling parameter is 20 in total. Simply comparing Fig. 5-3 with Fig. 5-4, we can understand that fewer parameters are used in the multi-pooling calculation. This is to say that network structure with multi-pooling process can carry out faster computation than that of pyramid pooling structure. In the multi-pooling structure, the small size pooling layer is derived from the pooling computation of the larger size pooling layer. This computation method provides the same output as SPP but uses fewer parameters.

After convolution, the features of all objects within the image are extracted, but these features still retain the distribution in various directions, which is represented as the distribution in the x - y direction in the two dimensional image. Parallel to multi-pooling layers, the calculation of PCA express these distribution features in a specific value for the two dimensional image. The distribution characteristics of the object in the x - y direction will remain unchanged even if the scale changes. This means that the distribution characteristic of the object is scale-invariant. In the following session, the reason why normalized eigenvalue calculated by PCA from feature map can be used as the feature of scale invariance is described.

5.3.3 Feature of scale invariance derived from PCA process

In this session, the reason of why the eigenvalues of feature map can be used as the feature of scale invariance for the same object at different scale is explained. In referring to Fig. 5-2, the PCA process and multi-pooling layers are decomposed for comprehensive description.

The output (Fig. 5-2②) from convolution layers is used as the input to PCA. The original image is input to the network and through convolutional calculations from conv_1 to conv_5, the background noise of the original image is filtered out and remained categorized object class consists of the feature map dataset by specific local

shapes which are activated by different convolutional filters. In the network flow, one filter can only activate one specific shape on the feature map. Besides, the feature map size is smaller than the original image size due to convolutional processing. The PCA application is at the pooling phase with small feature map size. The proper sliding window size is selected to cover different positions of the feature maps. The PCA process(Fig. 5-2 ③ ④ ⑤) consists of the steps of binarization, x-y position representation, covariance matrix, eigenvalue calculation and its normalization (Fig. 5-2⑤). Object shape is composed of pixel point positions. In 2-D image the object pixel points mainly spread along two orthogonal directions. The spread degree of pixel points of the same object at different scale is invariant. Eigenvalue of covariance matrix indicates the spread degree of the pixel points along two orthogonal directions. Because the eigenvalues are calculated based on variance, and the variance of image point distribution is calculated based on pixel distance. The eigenvalues will vary according to the size of the target object. The eigenvalues of an object will increase when its original size increases. Therefore, we need a method to normalize the eigenvalues to make them dimensionless in order to extract the feature of scale invariance.

Refer to below calculation formula Eq (5-1) for eigenvalue normalization:

$$n\lambda_k = \frac{\lambda_k}{\sqrt{\lambda_1^2 + \lambda_2^2}} \quad (5-1)$$

k : eigenvalue numbers, 1 or 2

$n\lambda$: dimensionless eigenvalue or normalized eigenvalue

λ : eigenvalue of covariance matrix

As a simple example in Fig. 5-5, an ellipse with radii of (2,1) and another ellipse with radii of (4,2) have eigenvalues of [2,1] and [4,2], respectively. The normalized eigenvalues of both ellipses are same as [0.8944, 0.4472]. This theoretic evidence proved that the same object at different scales has the equal normalized eigenvalues.

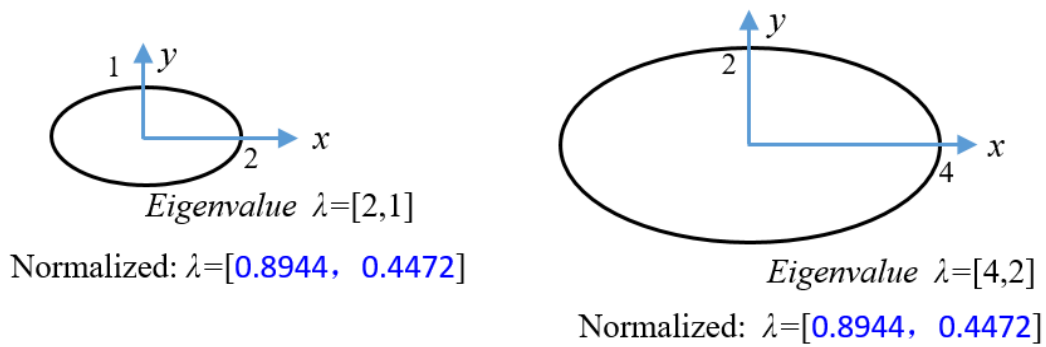


Figure 5-5 Eigenvalues of the same object at different scales

In order to validate the concept of using normalized eigenvalue as the feature of scale invariance, PCA calculation is performed to the binarized images where the original images are captured from real road traffic scene. The covariance matrix is composed of pixel position other than pixel RGB values. As the results, the similar normalized eigenvalues $n\lambda_1$ & $n\lambda_2$ ($n\lambda_1 \approx n\lambda_2$ in Fig. 5-6) can be obtained with regarding to the same object at different scale in the image.

With regarding to the proposed network structure in Fig. 5-2, PCA application is executed after conv5_3. This means the original image has been calculated through the designed convolutional filter iterations and each activated shape is evolved to the

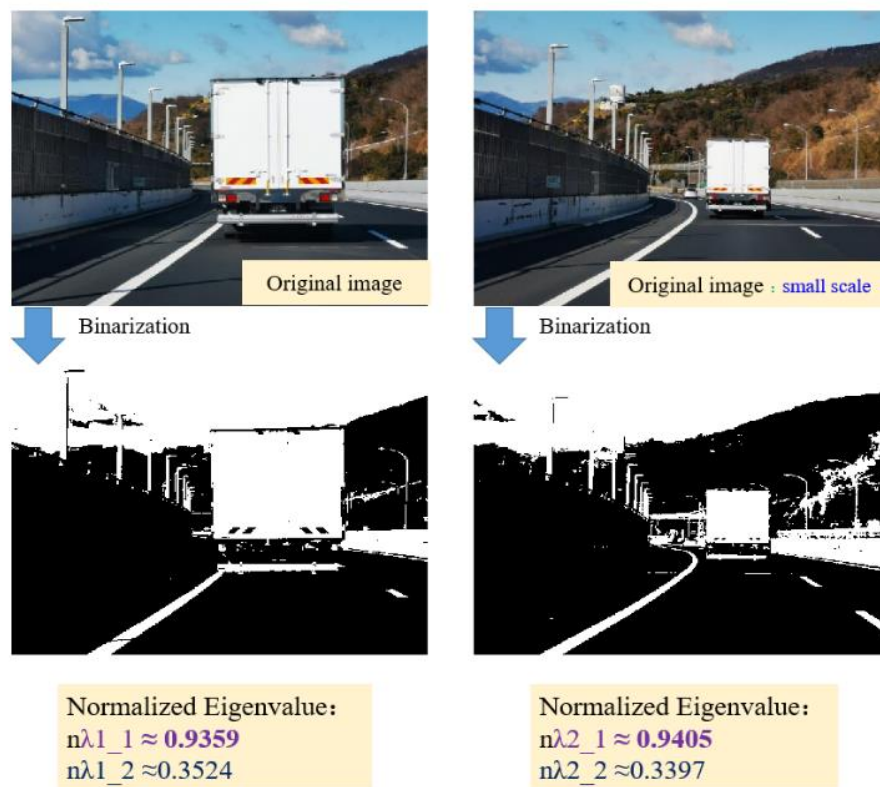


Figure 5-6 Normalized eigenvalues

feature map, referred to visualized feature in Fig. 5-7. When PCA is applied to each feature map, the normalized eigenvalues of different object scales become nearly equal because of the noise reduction by convolution operation as the filtering function. As the evidence of the proposed algorithm processing, some visual and binarized feature maps are selected as the convolutional outputs of original image, small scale object image and large-scale object image (Fig 5-7). These outputs are calculated through PCA processing and the results of normalized eigenvalues are very consistent ($\lambda_3 \approx \lambda_4 \approx \lambda_5$).

So, we can conclude that the normalized eigenvalues of PCA possess the feature of scale invariance and they are regardless to object scales. Based on this fact, the proposed algorithm can be implemented by using PCA process parallel loop as the support for the extraction of scale invariant feature for the same object at different scales, shown in Fig 5-2.

The eigenvector and eigenvalue obtained by PCA represent the distribution direction of the image and the magnitude of the object along the feature vector, respectively.

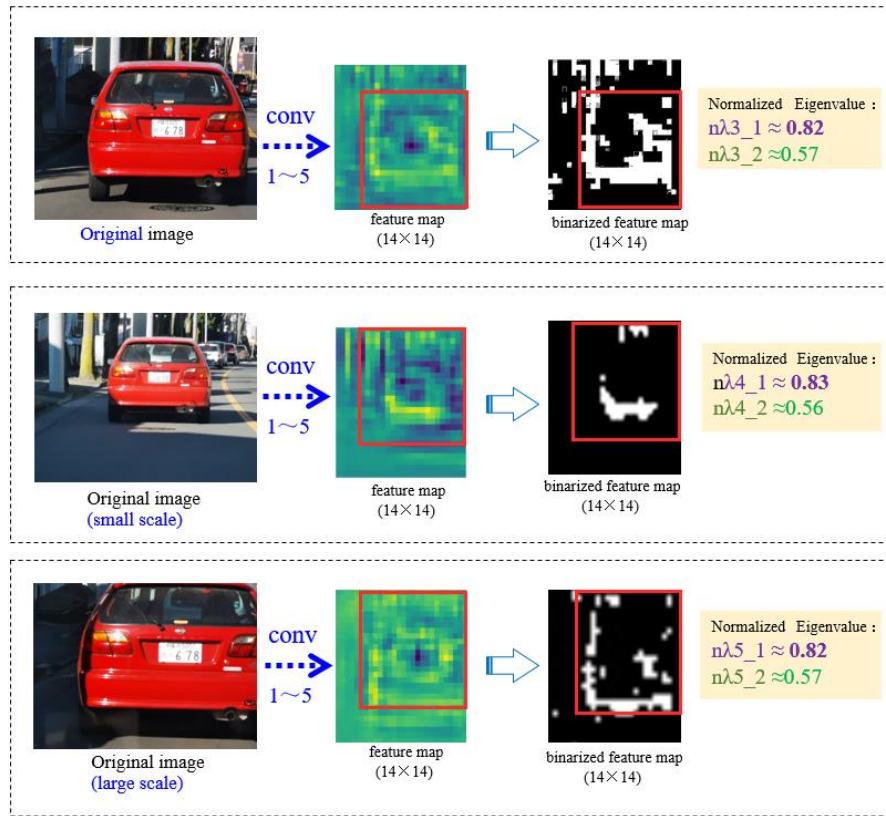


Figure 5-7 Demonstration of normalized eigenvalues

The binarized feature maps derived from the convolution layers are used as the input to PCA processing. The threshold of binarization is the mean value of the feature map matrix. After convolution, the image noise and the influence of non-target objects can be reduced. According to the heat map representation of the feature map, and the feature expression of the shape distribution, the shape of 2-D object is normally described by the axis with two variables. To analyze the shape distribution by PCA, the covariance of these two variables is computed. The feature maps are binarized and the stronger activated pixels are remained. We can transform the matrix of the feature map into a set of points located on the 2-D coordinate axis and use the coordinates of these points as calculation parameters. The formula is as follows:

The covariance $\sigma(x,y)$ of two coordinate variables x and y is given by Eq (5-2). The covariance matrix of the feature map is expressed by Eq (5-3).

$$\sigma(x,y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (5-2)$$

$$C = \begin{bmatrix} \sigma(x,x) & \sigma(x,y) \\ \sigma(y,x) & \sigma(y,y) \end{bmatrix} \quad (5-3)$$

And we can obtain the eigenvalue $\lambda=[\lambda_1, \lambda_2]$ and eigenvector $v=[v_1,v_2]$ from the covariance matrix C by Eq (5-4).

$$\lambda e = Ce \quad (5-4)$$

Then eigenvalues are normalized by Eq(5-1).

The PCA process flow is showed as below Fig. 5-8, with feature map size $p \times p$ and window size $m \times m$. The sliding window is used to shift over feature map for each filter activated area. As the calculation results, the normalized eigenvalues are sent as scale-invariant features to the fully connected layers (Fig. 5-2). Together with multi-pooling layer output to fully connected layers, the neural network calculation is performed accordingly. The detail network model structure is discussed in next session.

5.4 Network model verification for scale variant object recognition

5.5.1 Algorithm structure

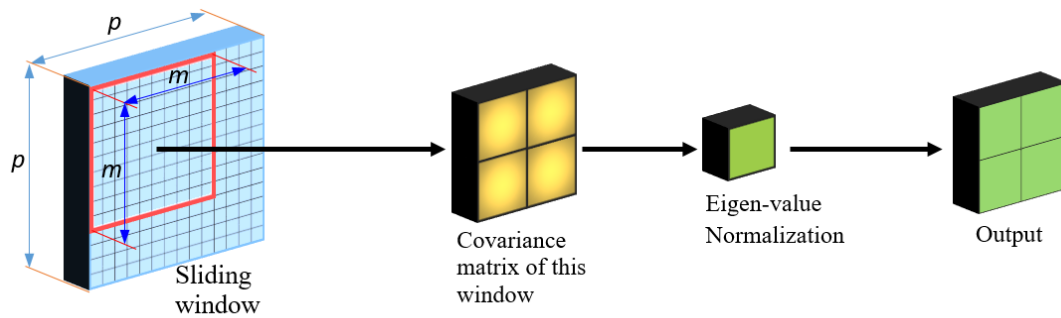


Figure 5-8 PCA process

The proposed network model structure is configured as Fig. 5-9. The parameters are selected as per VGG-16; but multi-pooling, PCA process and the first fully connection layers need to have different parameter setting, referred to Fig. 5-10.

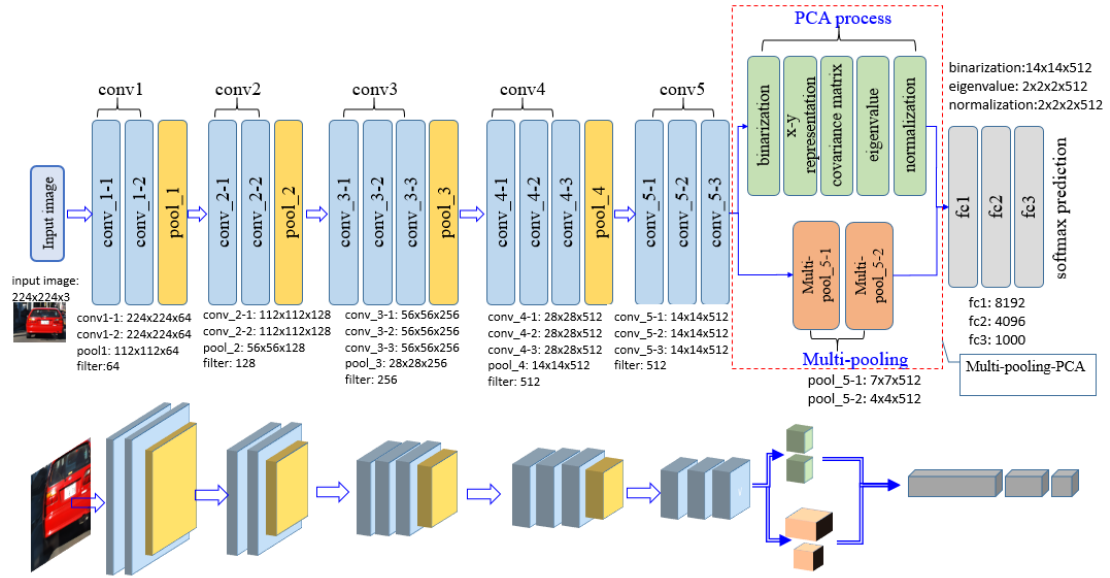


Figure 5-9 Proposed algorithm model

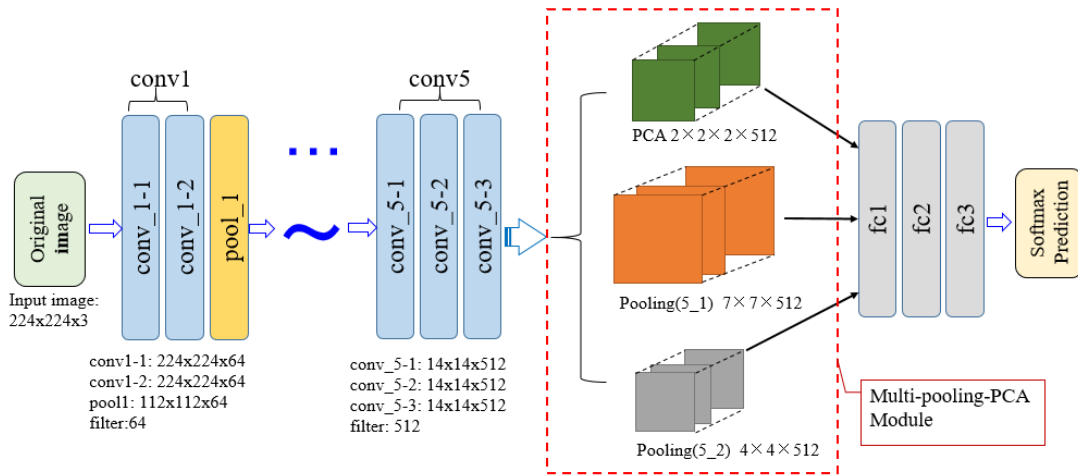


Figure 5-10 Parameters of algorithm model

Provided that the network input image size is $224 \times 224 \times 3$ (RGB), the iterating convolutional and pooling layer calculations are conducted from conv-1 to conv-5. In the network flow, one filter can only activate one specific shape on the feature map. 512 filters are used in the object shape activations as shown in Fig. 5-9. The output size of CNN calculations from conv_1 to conv_5 is 14×14 , and the sliding window size is chosen as 10×10 to cover different positions. The PCA application is at the pooling phase with the sliding window size which is partially covering the feature map. The object scale invariance features are extracted from PCA process. The loss of cross entropy is evaluated with the network output. The loss formulation is defined as Eq (5-5):

$$L = -\sum_i^n y_i \ln(a_i) \quad (5-5)$$

where i: sample; y: label; a: output; n: total number of the samples.

Accuracy *Acc* is taken as the accurate ratio of correct predictions, and its definition is Eq (5-6):

$$Acc = \frac{N_c}{B_s} \quad (5-6)$$

where *N_c*: number of correct predictions *B_s* : batch size.

5.5.2 Model training and testing

There has not been a public dataset existing for the purpose of scale variant object detection. Therefore, we are motivated to setup the dataset by the road vehicle driving scenario as the safety surveillance requires to detect the road vehicle at different distance. That means the vehicle scale is different within the camera captured image. Such vehicle image dataset-1 includes 500 sample images in our study. 400 sample images are taken for model training, and part of the demonstrated example images is shown in Fig. 5-11. Except for the 400 training sample image data, other 100 sample

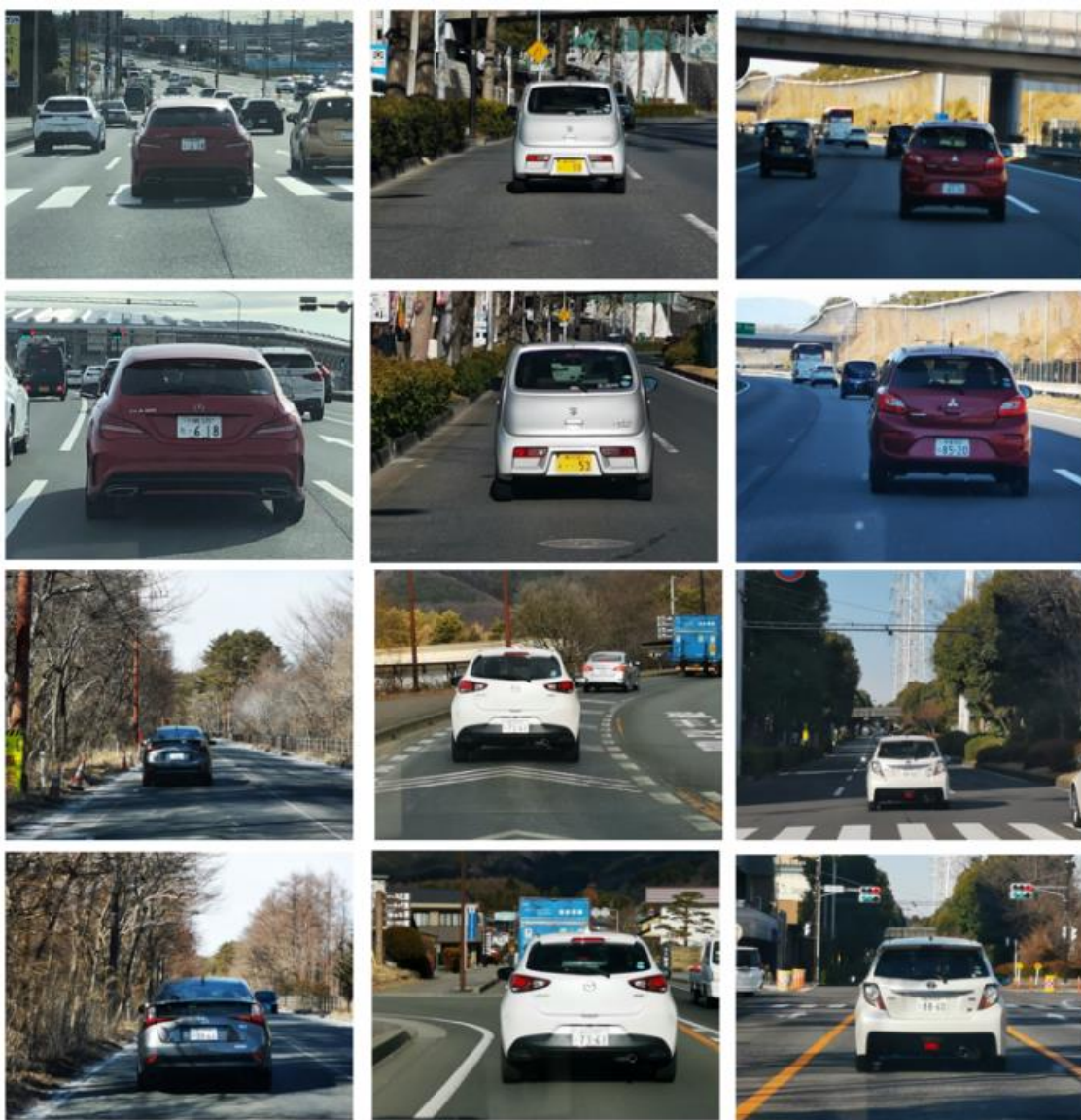


Figure 5-11 Training image data demonstration for variant scale vehicle object images (part of 400 image dataset)

images are used for testing, which are shown in Fig. 5-12. The calculations are done with batch size 16 and optimizer of stochastic gradient descent (SGD).

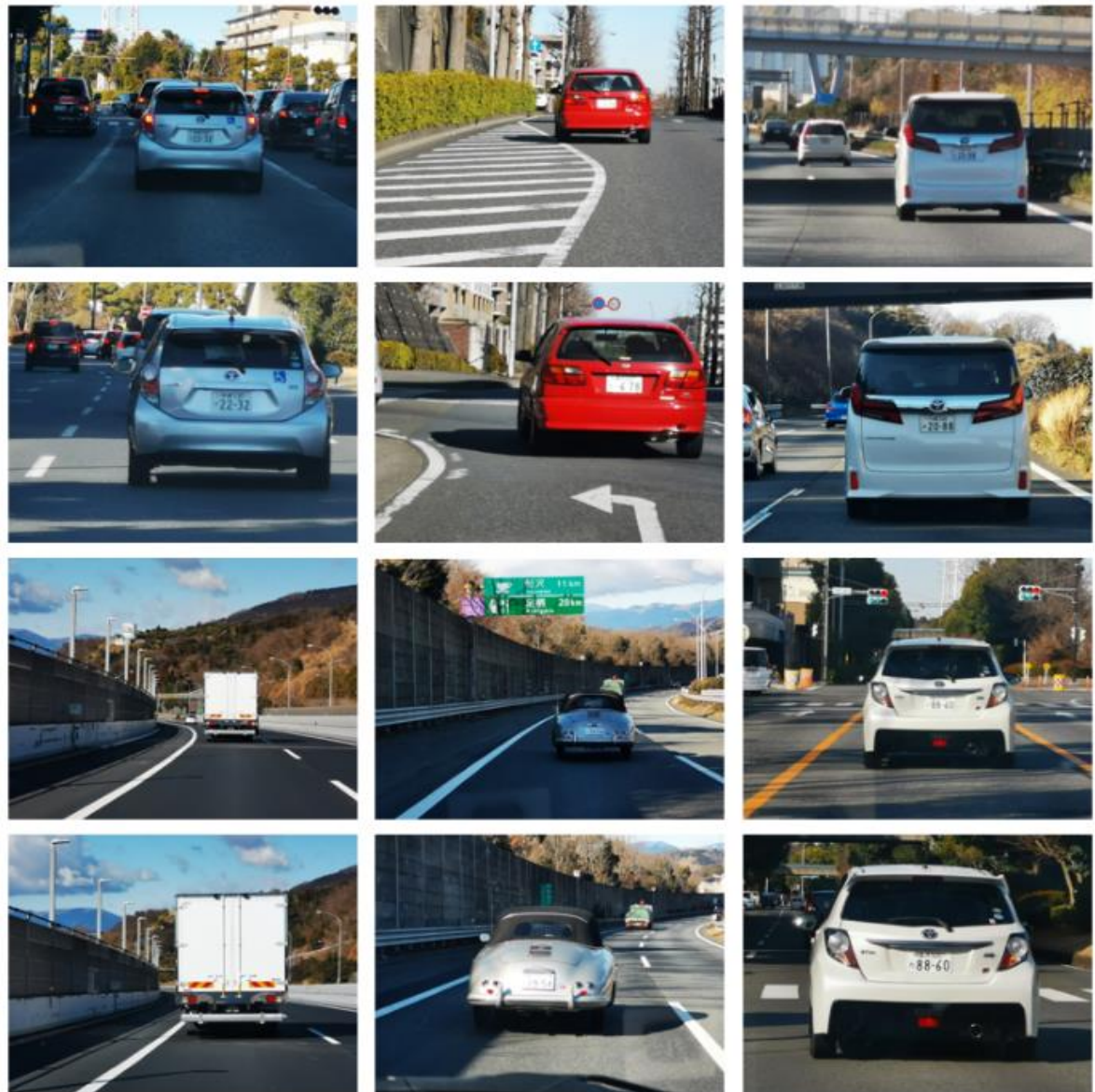


Figure 5-12 Testing image data demonstration for variant scale vehicle object images (part of 100 testing samples)

In order to compare the effectiveness of the proposed method, VGG-16 net and VGG-16 with spatial pyramid pooling network models are also trained and tested by using the above-mentioned sample images. The outputs in the experiment of dataset-1 are ‘vehicle’ and ‘the other thing’.

Fig. 5-13 indicates that the trained VGG-16 network model is applied to the testing sample images of variant object scales and reached to the computational convergence with loss factor $L \approx 0.904833$, and average testing accuracy $Acc \approx 0.7919$.

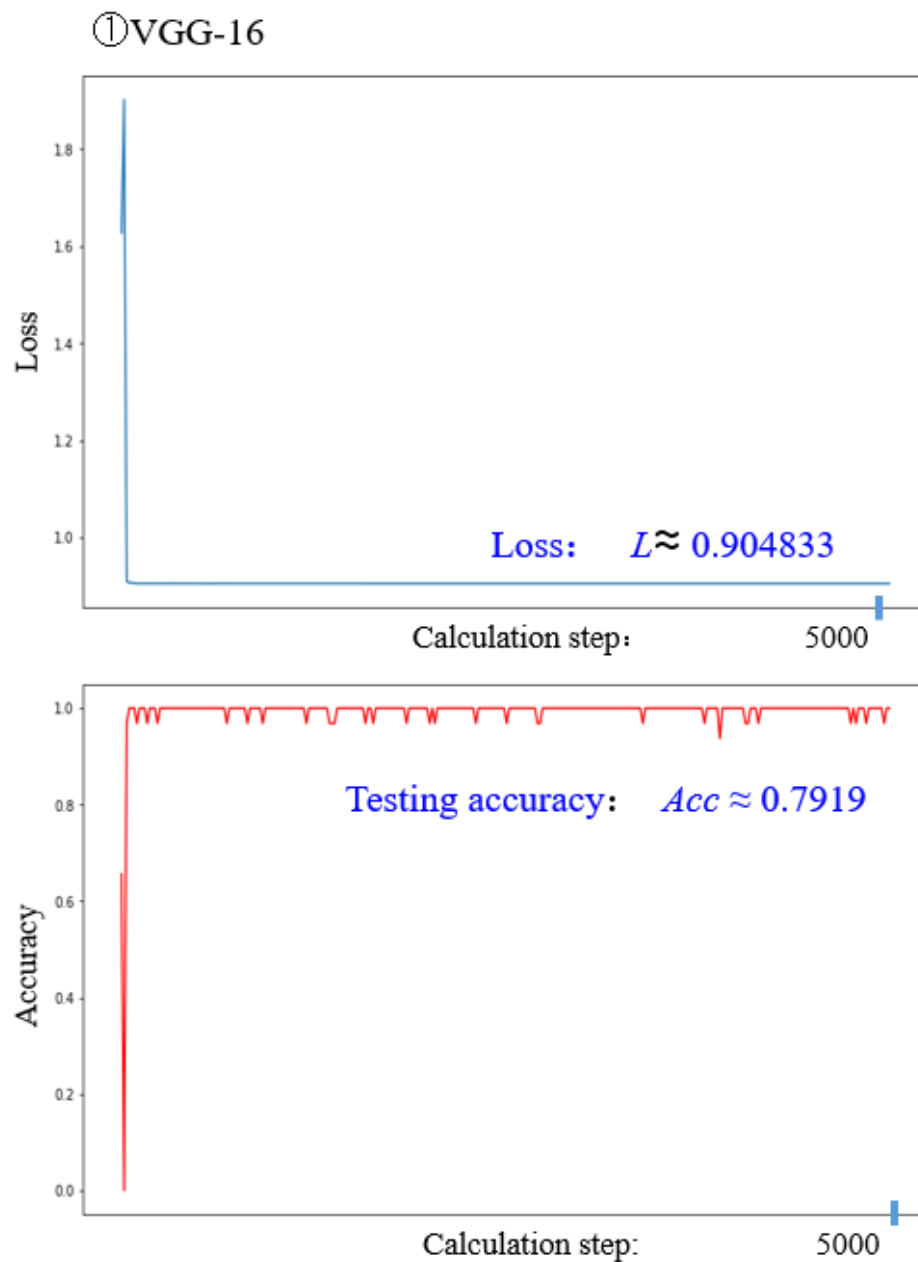


Figure 5-13 VGG-16 network performance

Fig. 5-14 indicates that the trained VGG-16 & SPP network model is applied to the testing sample images of variant object scales and reached to the computational convergence with loss factor $L \approx 0.904834$, and average testing accuracy $Acc \approx 0.8229$.

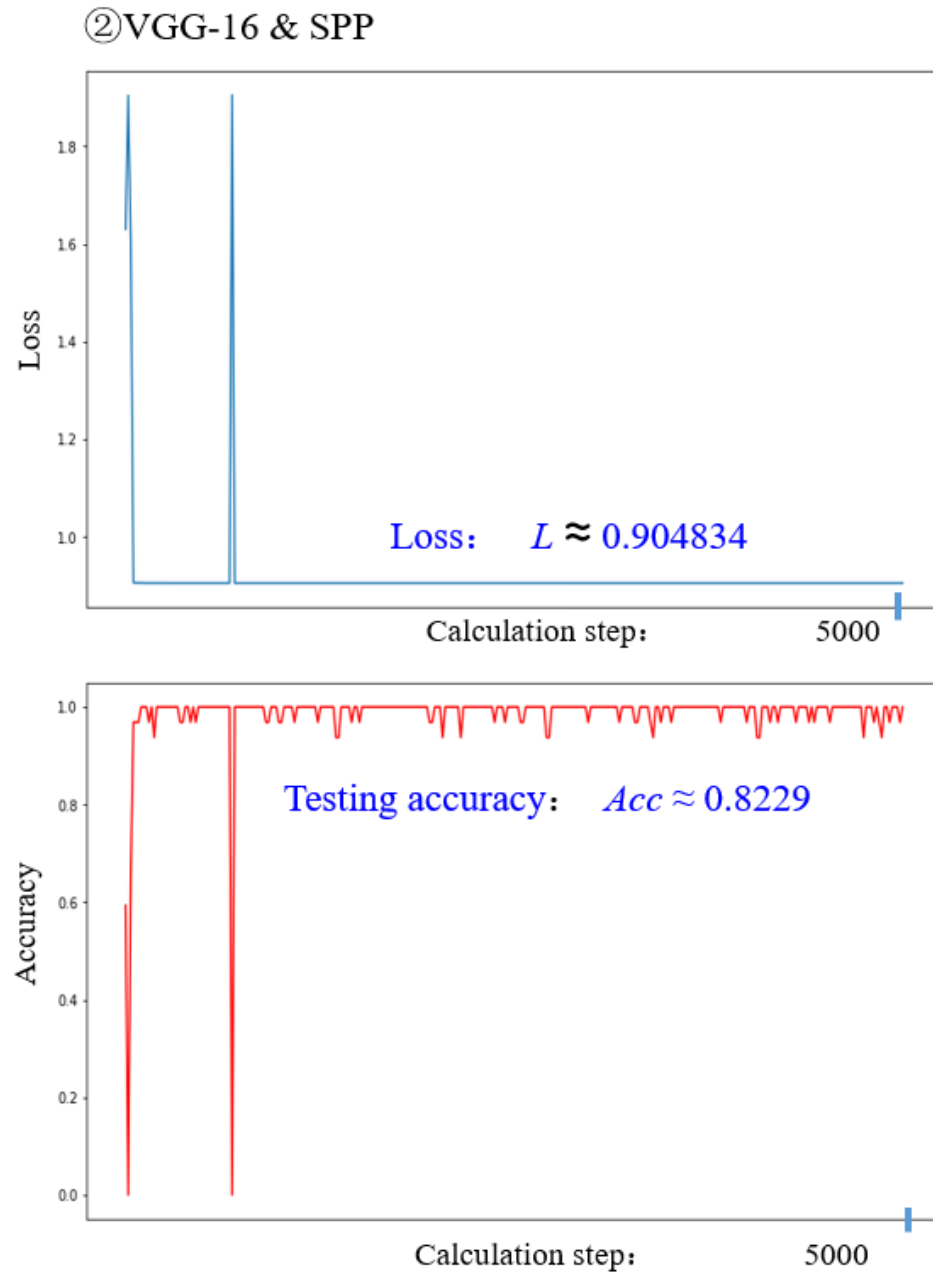


Figure 5-14 VGG-16 & SPP network performance

Fig. 5-15 indicates that the trained VGG-16 & proposed network model is applied to the testing sample images of variant object scales and reached to the computational convergence with loss factor $L \approx 0.909254$, and average testing accuracy $Acc \approx 0.8434$.

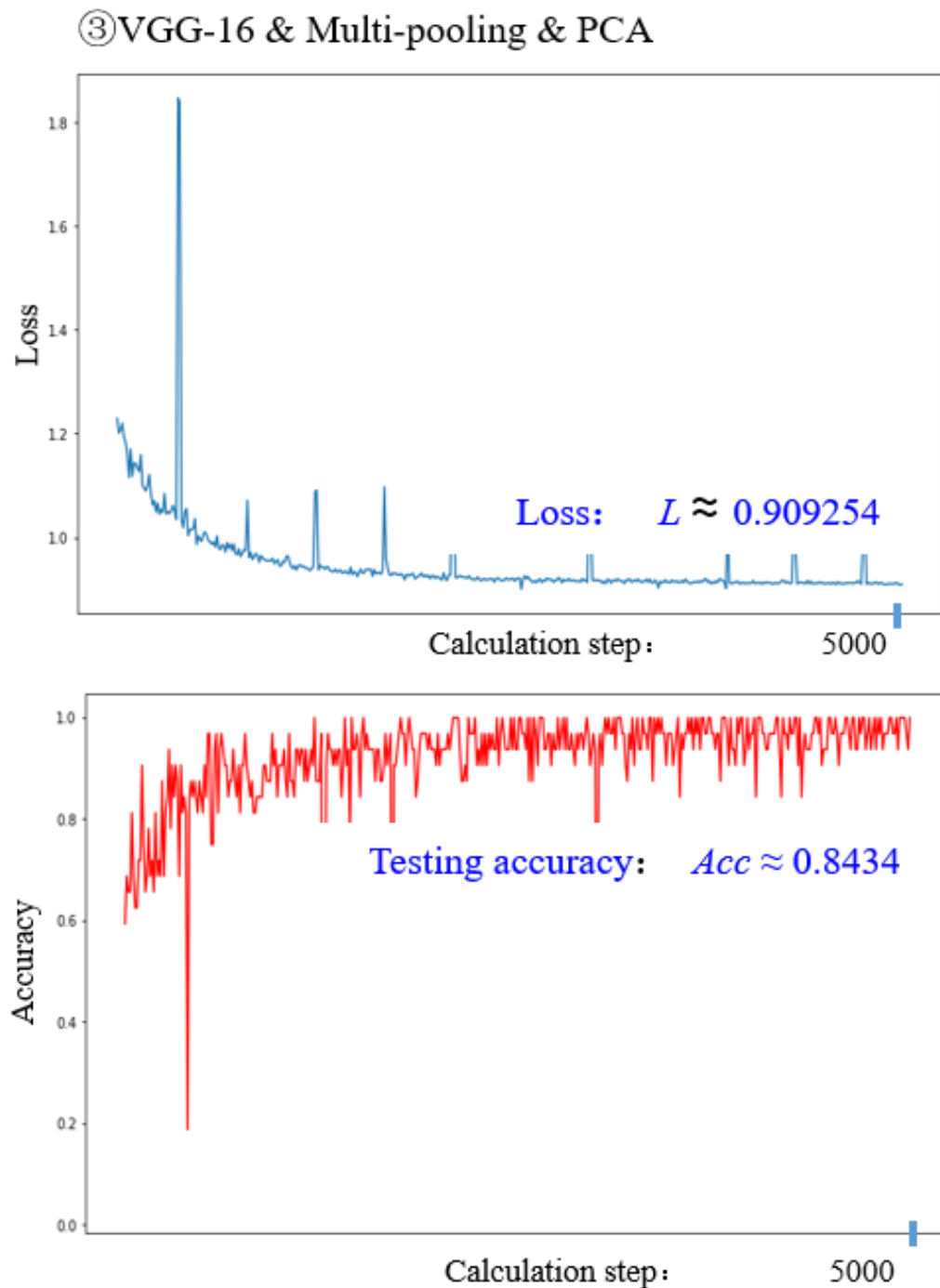


Figure 5-15 VGG-16 & proposed network performance

The summary of these networks which are embedded with different modules is listed in Table 5-1. The proposed network ③VGG-16 & Multi-pooling & PCA shows the best accuracy results.

Table 5-1(a) Performance comparisons of 3 network testing results

Network	①VGG-16	②VGG-16 & SPP	③VGG-16 & Multi-pooling & PCA
Accuracy	0.7917	0.8229	0.8438

Table 5-1(b) Confusion matrix

①VGG-16			②VGG-16 & SPP			③VGG-16 & Multi-pooling & PCA		
Output	“Car” class quantity	“Other” class quantity	Output	“Car” class quantity	“Other” class quantity	Output	“Car” class quantity	“Other” class quantity
“Car”	66	6	“Car”	70	5	“Car”	69	6
“Other”	14	10	“Other”	12	9	“Other”	9	12
Sum	80	16	Sum	82	14	Sum	78	18
True Class	66+10=76	76/96= 0.7917	True Class	70+9=79	79/96= 0.8229	True Class	69+12=81	76/96= 0.8438

Table 5-1(b) shows the prediction accuracy of each object category. Testing dataset is shown in figure 5-12. 96 samples are tested by each structure.

In order to prove that the proposed method is valid with regarding to other category objects, we use the scale variant and different category objects, e.g. cars, cats, dogs, flowers as the training sample image dataset-2, referred to Fig. 5-16.

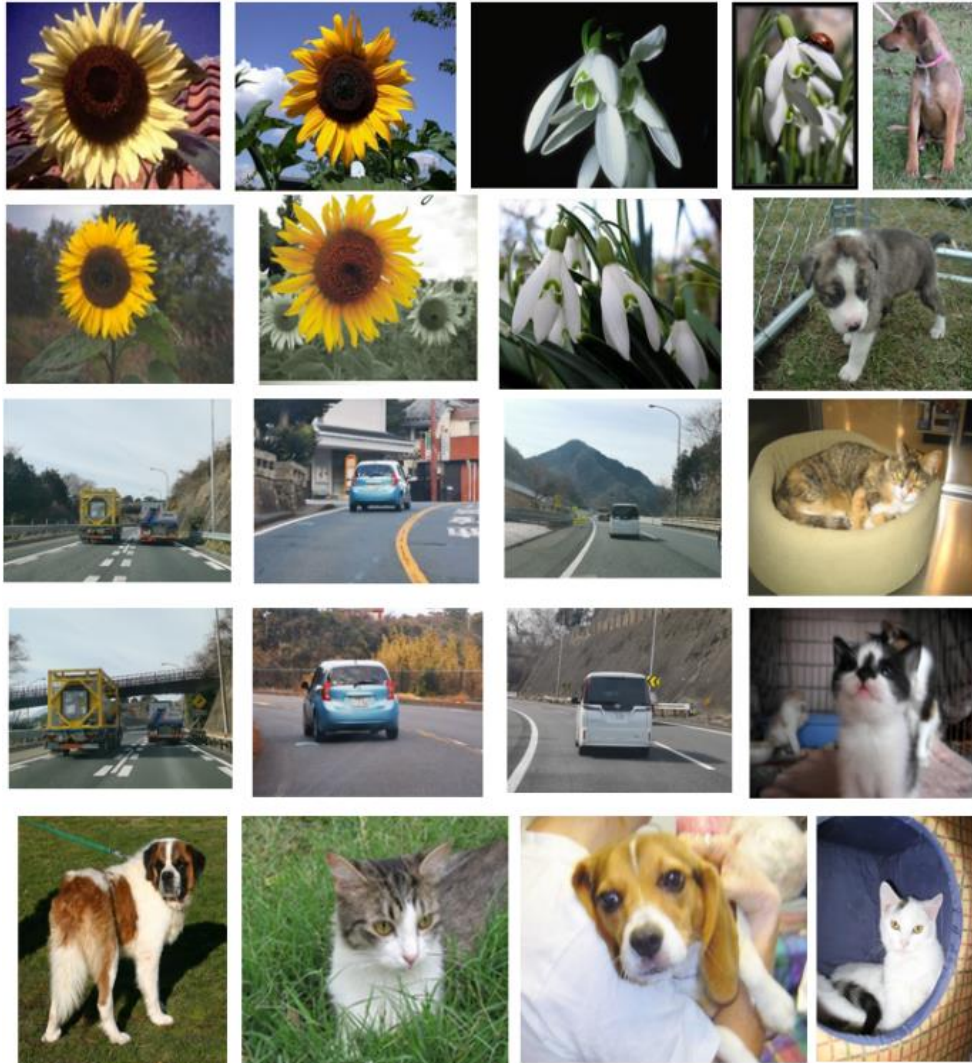


Figure 5-16 Object recognition of different object category in different scales

Dataset-2 has 600 sample image data which include five category image data. The categories are sun flowers, white flowers, cats and dogs as well vehicles, with each category 100 images and total 500 images for network model training. 100 images different from the total 500 sample images are used to make the testing with each category 20 sample images. The outputs are ‘vehicle’, ‘cat’, ‘dog’, ‘white flower’, ‘sun flower’ and ‘the other thing’. The results (shown in Fig. 5-17) demonstrated the proposed network performance for different single object category in different scales. The computational convergence loss is $L \approx 0.67$ and the obtained average accuracy is 0.8802.

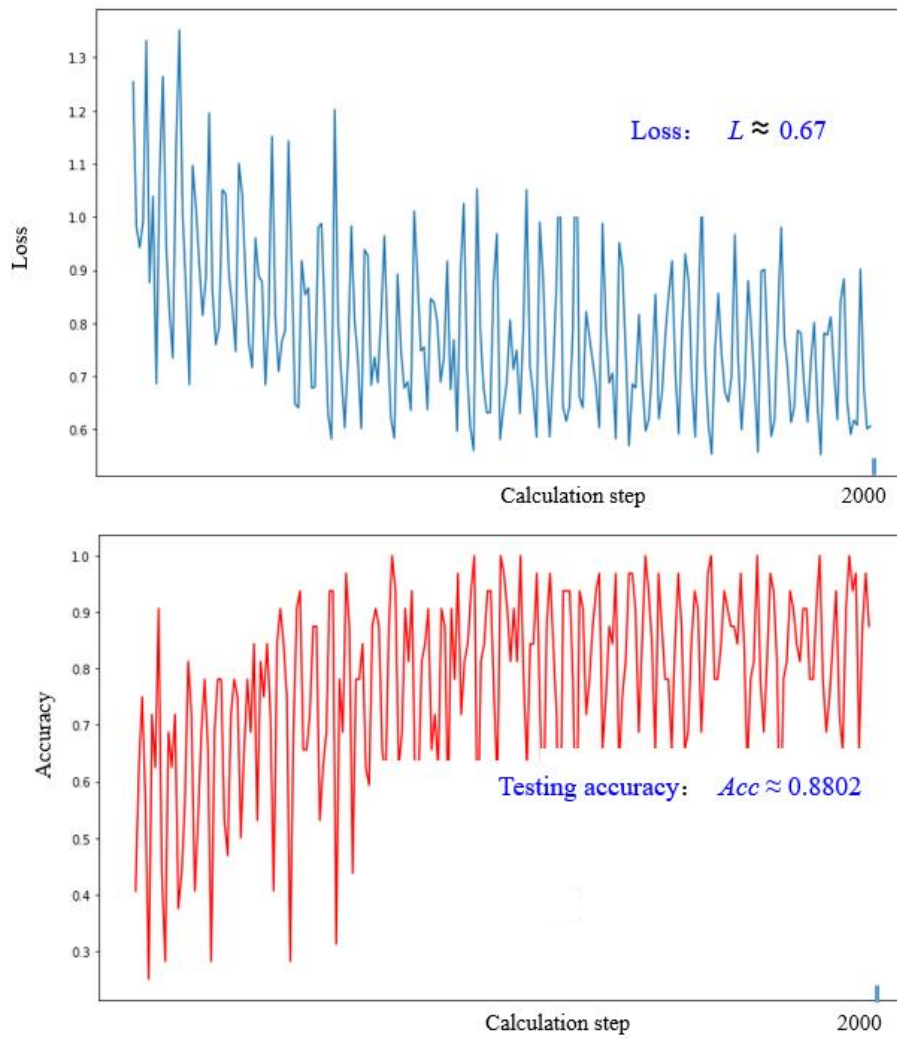


Figure 5-17 Proposed network performance for object detection of different single object category at different scales

5.5.3 Network performance comparisons

(1) Feature Selective Anchor-Free (FSAF)^[72] is a network which is focus on the scale variation of objects. The overview of FSAF module is shown in the Fig.5-18.

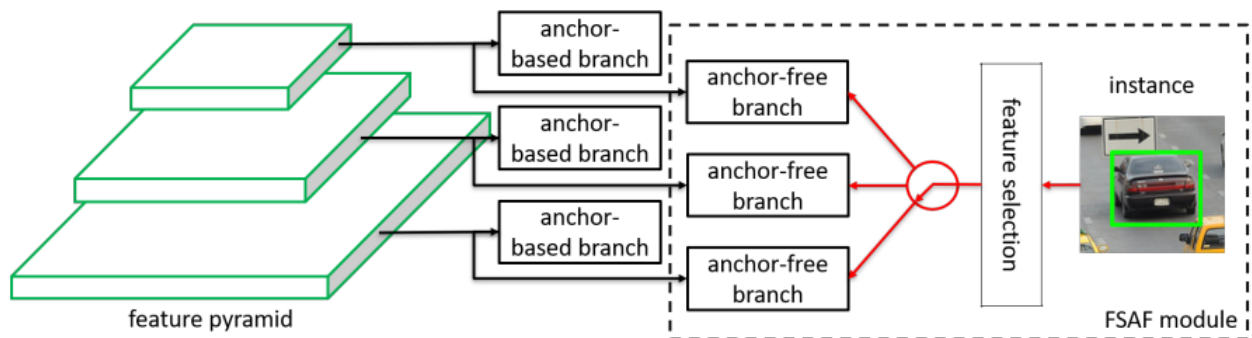


Figure 5-18 FSAF module plugged into conventional anchor-based detection method

FSAF prediction: no cars on the image:



The proposed method:



Figure 5-19 Comparison of FSAF module and the proposal

During training, each instance is assigned to a pyramid level via feature selection for setting up supervision signals. FSAF is a feature selectivity and anchor-free design, combined with hierarchical prediction and classification-regression branches, to achieve accurate and efficient object detection. This network is applied to test the vehicle image in dataset-1. 100 vehicle images are tested. If a car is successfully detected in the image, the prediction is counted as correct. And the final correct prediction ratio of tested vehicle images is 0.74.

Fig.5-19 shows the comparison of vehicle detection by FSAF and the proposed method.

(2) Besides of CNN networks, SIFT^[73] is applied to match vehicles on road to compare the performance of proposed structure. In this SIFT application, 100 tested samples are from the dataset-1. For classification task on the dataset-1, SIFT+K-means method^[74] is used for classification about whether the image belongs to vehicle category or belongs to the other thing. K-Means clustering algorithm is a typical unsupervised learning algorithm, which is mainly used to automatically group similar samples into one category. In the K-Means clustering algorithm, samples are divided into different categories according to the similarity between samples.

Fig. 5-20 shows the matching key-points derived from SIFT calculation. The accuracy is defined as the ratio of the correct classification of SIFT+K-means. The result



(a) examples of correct SIFT matching results



(b) examples of failed SIFT matching results

Figure 20 examples of SIFT matching results

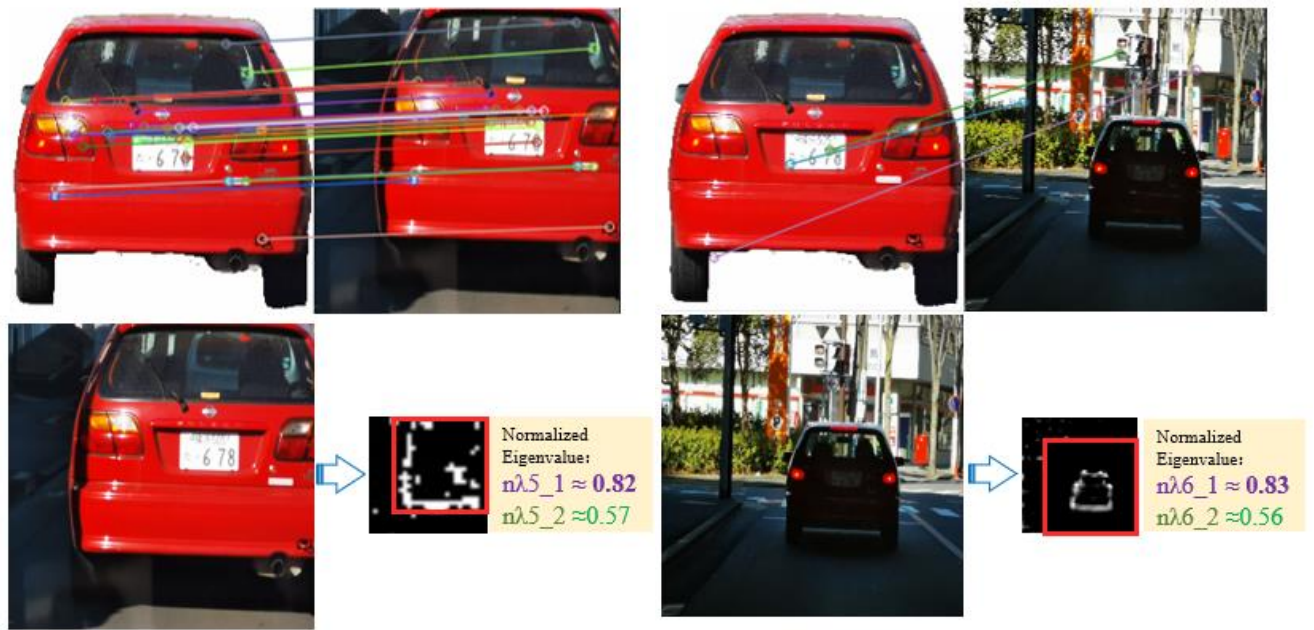


Figure 5-21 comparison of SIFT and the normalized eigenvalues

accuracy of SIFT+K-means classification is **0.65**. The reason of SIFT performs low accuracy in vehicle detection is that SIFT is sensitive to the lighting change and it is limited to the large range scale variation which is shown in fig.5-20(b). The lighting condition is always changing outdoor and the accuracy of SIFT becomes low accordingly. Fig.5-21 shows the comparison of SIFT and the normalized eigenvalue in the lighting change and scale variant conditions. The normalized eigenvalues keep the same but the matching key-points are missing in the SIFT calculation. Table 5-2 shows the comparison of SIFT, FSAF, VGG16, SPP module and Multi-pooling module our proposed approach. The accuracy of SPP and Multi-pooling are the same. But the computation time with own PC of SPP is 148.5ms for one image on average, which is 9.5ms more than Multi-pooling (139ms on average).

Table 5-2 comparison of SIFT, FSAF, VGG16, SPP module Multi-pooling module and the proposal.

Network	SIFT+ Kmeans	FSAF	VGG-16	Spp	Multi-pooling	Network module of multi-pooling & PCA
Accuracy	0.65	0.74	0.7917	0.8229	0.8229	0.8438

Note: Dataset-1 with scale variant object classification

(3) To verify the proposed network structure of VGG-16 & Multi-pooling & PCA, 5 networks are tested by 100 sample images from the above described dataset-1 with the classification of only scale variant vehicles. Network ViT-L/16^[75](2023), Nasnet-A(6)^[76](2018) and Resnet-152^[77](2016) as well as VGG-16^[40](2015) reflect the state-of-

the-art techniques. The proposed method achieves relative accurate testing result $Acc=0.8846$ with the classification of only scale variant vehicles, shown in Table 5-3.

Moreover, by using dataset-2 with the classification of scale variant object categories (sun flowers, white flowers, cats, dogs, vehicles) and other things, five

Table 5-3 Performance comparisons of 5 network testing results

Network	ViT-L/16	Nasnet-A(6)	Resnet-152	VGG-16	Network module of multi-pooling & PCA
Accuracy	0.6923	0.8000	0.7185	0.8250	0.8846

Note: Dataset-1 with scale variant vehicle classification

networks are also tested by 100 sample images from the above described dataset-2. The proposed method achieves comparable testing result $Acc=0.8802$, shown in Table 5-4. From the cost-performance point of view, as VGG-16 is the base, the proposed network can achieve the nearly same accuracy result as the latest network ViT-L/16^[74](2023) with regarding to the scale variant object recognition.

Table 5- 4 Performance comparisons of 5 network testing results

Network	ViT-L/16	Nasnet-A(6)	Resnet-152	VGG-16	Network module of multi-pooling & PCA
Accuracy	0.8900	0.8210	0.8263	0.7708	0.8802

Note: Dataset-2 with five categories, scale variant object classification

The image classification performance of the proposed network (VGG-16&Multi-Pooling&PCA) is also experimented on the public dataset ImageNet. ImageNet includes 1000 categories of objects, 15 million images and 22 billion parameters. The comparison result of the proposed network and the other networks are shown in the Table 5-5.

Table 5- 5 Performance comparisons of 5 network testing results

Network	ViT-L/16	Nasnet-A(6)	Resnet-152	VGG-16	Network module of multi-pooling & PCA
Accuracy	0.8960	0.8270	0.7857	0.7440	0.8470

Note: Public dataset ImageNet

(4) To verify the applicability of proposed network module for the task of object detection, the proposed module is implemented in the YOLO network. The proposed module is embedded before the fully connected layer of YOLO as Fig. 5-22 shown.

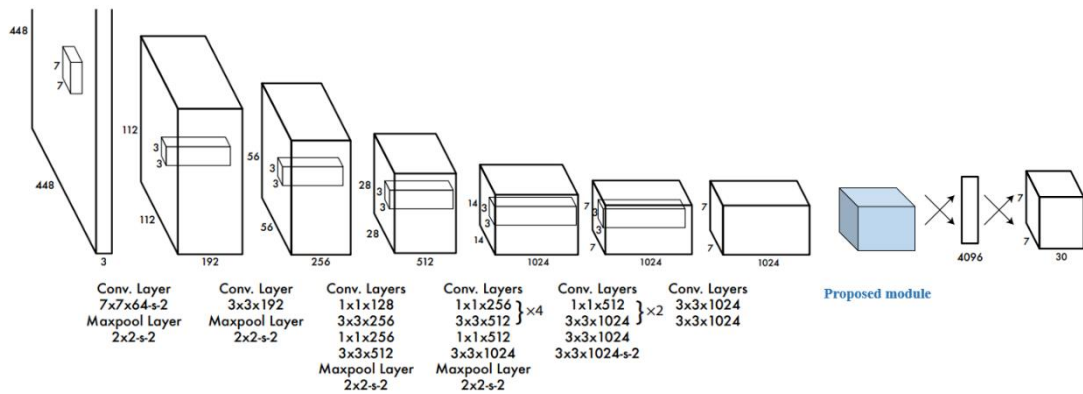


Figure 5-22 proposed module embedded in the YOLO

To evaluate the proposed module, the above network structure (Fig. 5-22) is trained and tested on the public dataset PASCAL VOC2007+2012. This dataset examples are shown in fig. 5-23



Figure 5-23 PASCAL VOC dataset

Fig. 5-24 shows the detection anchor of the proposed module. The average precision (AP) is used for evaluating the performance of detection. The AP of YOLO on the dataset of PASCAL2007+2012 car class is 0.559, and mean average precision (mAP) is 0.579. The AP of our proposed module on the PASCAL2007+2012 car class is 0.621, and mAP is 0.627.

The result proves that the proposed network can enhance the detection network.



Figure 5-24 detection result examples of the proposed module

5-5. Discussions

For the purpose of object recognition at different scales, the single entire object image is input to convolutional network and the feature map for PCA input is derived after 5 repeated convolutional and pooling layer calculations. This process is equal to apply noise filter to the image data, so we can expect the feature map data is more focused on the activated object. In this case, the feature of scale invariance is embedded through the module of parallel PCA processing with multi-pooling layers, and then fully connected layers produce the Softmax accuracy prediction for scale variant object recognition.

According to the above verification of the proposed network, the effective and robust detections for the scale variant objects are determined, especially for traffic vehicles which are scale variant due to moving status and changeable viewpoint distances. Our proposed network module structure achieved the SOTA result. The algorithm performance and robustness improvement is the eternal topic especially in automotive engineering field which is required as per SOTIF (Safety of the intended functionality) international standard^[78]. The proposed method is verified, though vehicle testing image samples are currently not big enough. Much more real-world

sample image dataset is necessary for a reliable algorithm application in product development.

5-6. Summary

The moving object presents different scales in the captured image due to the perspective effect of different viewpoint distances. The recognition of the same object at different scales requires feature learning with scale invariance. The normalized PCA eigenvalues calculated from feature maps are used to extract scale-invariant features. We proposed a CNN structure embedded with the module of multi-pooling-PCA for scale variant object recognition. Multi-pooling layers make use of fewer parameters in computation and PCA process contributes to the feature extraction of scale invariance. The validation of the proposed network structure is verified by scale variant vehicle image dataset, because in the real traffic situation, the moving vehicle appears different scale in the video camera image due to dynamic observing distance. Compared with scale invariant network algorithms of SIFT and FSAF as well as miscellaneous networks, the proposed network can achieve the best recognition result. To testify the practicality of this proposed network structure, the testing of public dataset ImageNet is conducted and the comparable results proved its effectiveness in general purpose of applications. Moreover, to verify the applicability of the proposed module for the task of object detection, the module is implemented into YOLO network and is located before the fully connected layers of YOLO. The testing result proves that the modified YOLO network by the proposed module can achieve better result than YOLO network itself. The testing results of these networks demonstrated that the proposed method can achieve the comparable detection accuracy according to the SOTA result. The consequent expression is that the proposed module can be embedded to other deep CNN to enhance the network performance and robustness.

Chapter 6. Conclusion remarks

Inferring depth information from 2-D image is a cost-effective method to obtain 3-D depth information. A lot of techniques for depth feature extraction have been developed. Our new approach is to make use of the texture cues around focus area and the blur cues nearby camera. Through the comparison of texture cue and blur cue, the depth cue can be extracted to indicate the distance classification information. By using principal component analysis, the object edge lines are extracted, and their spatial frequency density is calculated by discrete cosine transform. Blur cues are only calculated by discrete cosine transform. CNN performs the depth information learning in which image pyramid and coarse-to-fine approaches are utilized to reflect the spatial frequency variation information. The proposed method is verified by vehicle functional application of blind spot detection.

Object can be presented at variant scale due to deep variation. The smaller the depth is, the larger the scale will be; vice versa. Therefore, it is necessary to consider the change of object scale when CNN recognizes the object from 2-D image. As the novel approach, according to the basic mathematical derivation and practical verification, we proposed to implement the scale invariant feature extraction by the normalized eigenvalues of object feature map covariance analysis. And another new approach is the multi-pooling, which uses fewer parameters than spatial pyramid pooling in the calculation. As a novel structure module of parallel routine of principal component analysis and multi-pooling, its effectiveness is validated by our own dataset of scale variant vehicle object in the traffic situation, and also public ImageNet dataset. The start-of-the-art classification results are achieved in comparing with other latest networks. The proposed module can be embedded to YOLO network for object detection purpose to enhance the performance. The proposed network structure module can be embedded to the other deep convolutional neural networks to enhance the robustness.

There is no such a model that can cope with every situation in feature learning by deep convolutional neural network. The continuous efforts are required to make progress on the algorithm improvements.

References

- [1] Ko, J., Kim, M. and Kim, C., 2D-to-3D Stereoscopic Conversion: Depth-Map Estimation in a 2D Single-View Image, *Proceedings of SPIE* Vol. 6696, 2007
- [2] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng, Learning Depth from Single Monocular Images, *Computer Science Department, Stanford University*, Stanford, CA 94305
- [3] Wei, Q., Converting 2D to 3D: A Survey, *Information and Communication Theory Group, Delft Univ. of Technology*, the Netherlands, 2005
- [4] Liu, B., Gould, S. and Koller, D., "Single Image Depth Estimation From Predicted Semantic Labels", *the 23rd IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, San Francisco, CA, USA, 13-18 June 2010
- [5] Huang, X., Wang, L., Huang, J., et al., A Depth Extraction Method Based On Motion and Geometry for 2D to 3D Conversion, *Third International Symposium on Intelligent Information Technology Application*, 2009, pp.294-298.
- [6] Tsai Tsung-Han and Fan Chen-Shuo, Monocular vision-based depth map extraction method for 2D to 3D video conversion, *EURASIP Journal on Image and Video Processing* (2016)
- [7] Han Kyuseo and Hong Kihyun, Geometric and Texture Cue Based Depth-map Estimation for 2D-3D image conversion, *2011 IEEE International Conference on Consumer Electronics (ICCE)*
- [8] Hazirbas, C., Leal-Taix'e L. and Cremers, D., Deep Depth From Focus, *Computer Vision and Pattern Recognition*, April 2017
- [9] Mather George, "The use of image blur as a depth cue", *Perception*, 1997, volume 26, pages 1147-1158
- [10] Liu, L., Ouyang W., Wang X. Fieguth, P. Chen, J., Liu, X., Pietikäinen, M., Deep Learning for Generic Object Detection: A Survey, *International Journal of Computer Vision* 128:261–318 (2020) <https://doi.org/10.1007/s11263-019-01247-4>
- [11] Zou, Z., Shi, Wen, Guo, Y and Ye, J., Object Detection in 20 Years: A Survey, arXiv:1905.05055v2 [cs.CV] 16 May (2019)
- [12] Xiao L., Bahri, Sohl-Dickstein, Y., Schoenholz, Y, Pennington, J., Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks, arXiv:1806.05393v2 [stat.ML] 10 Jul 2018
- [13] Guo, Y., Kumazawa, I. & Kaku, C.: Blind Spot Obstacle Detection from Monocular Camera Images with Depth Cues Extracted by CNN. *Automot. Innov.* **1**, 362–373 (2018). <https://doi.org/10.1007/s42154-018-0036-6>
- [14] Guo, Y., Kumazawa, I., and Kaku, C.: Application Study of Blind Spot Monitoring System Realized by Monocular Camera with CNN Depth Cues Extraction Approach. *SAE Intl. J CAV* 2(4):219-232, (2019). <https://doi.org/10.4271/12-02-04-0016>.
- [15] Sánchez, R., Newman, D., A Primer on Complex Systems ISSN 0075-8450 ISSN 1616-6361, p103, Springer (2018)

- [16] Kong, T., Yao, A., Chen, Y., & Sun, F., HyperNet: Towards accurate region proposal generation and joint object detection. In CVPR (pp. 845–853) (2016). arXiv:1604.00600 [cs.CV]
- [17] Jolliffe, I.T. Principal Component Analysis, p24, ISBN 0-387-95442-2 2002 Springer-Yerlag New York, Inc.
- [18] Hua, J., Wang, J., Peng, H, et al.: A novel edge detection method based on PCA. Int. J.Adv.Comput. Technol. 3(3), 228–238 (2011)
- [19] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. IEEE Conference on Computer Vision and Pattern Recognition. 1(2), 2117-2125(2017).
- [20] Chen, Liang-Chieh, et al., Rethinking atrous convolution for semantic image segmentation, arXiv:1706.05587, 2017
- [21] Zhao, H, etc., Pyramid Scene Parsing Network, Proceedings of the IEEE Conference on CVPR, 2017, pp. 2881-2890
- [22] Eigen D. and Fergus, R. Predicting depth, surface normal and semantic labels with a common multi-scale convolutional architecture. arXiv:1411.4734, 2014.
- [23] Pinheiro, P., and Collobert, R., Recurrent convolutional neural networks for scene labeling. In ICML, 2014.
- [24] Lin, G., Shen, C., Reid, I. et al. Efficient piecewise training of deep structured models for semantic segmentation. arXiv:1504.01013, 2015.
- [25] Chen L., Yang, Y., Wang, J., Xu W, and Yuille A., Attention to scale: Scale-aware semantic image segmentation. In CVPR, 2016.
- [26] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A., Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv:1606.00915, 2016.
- [27] Singh B., and Davis, L., An analysis of scale invariance in object detection–snip, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3578–3587.
- [28] Cai, Z., Fan, Q., Feris, R., and N. Vasconcelos, N., A unified multi-scale deep convolutional neural network for fast object detection, in European Conference on Computer Vision. Springer, 2016, pp. 354–370
- [29] Berger T., Lettner D., al, What is a Feature? A Qualitative Study of Features in Industrial Software Product Lines, ISBN 978-1-4503-3613-0/15/07. 2015, DOI: <http://dx.doi.org/10.1145/2791060.2791108>
- [30] Lee, J. and Muthig, D., Feature-oriented Variability Management in Product Line Engineering. *Commun. ACM*, 49(12):55--59, Dec. 2006.
- [31] Lee, K., Kang, K.C., Koh, E., Chae, W., Kim, B. and Choi, B.W., Domain-Oriented Engineering of Elevator Control Software: A Product Line Practice. In *Proc. SPLC*, 2000.
- [32] Athiwaratkun B. and Kang K., Feature Representation in Convolutional Neural Networks[J]. *Computer Science*, 32(1):41-45. 2015

- [33] Razavian, A.S., Azizpour, H., Sullivan, J. and Carlsson, S., CNN features off-the-shelf: an astounding baseline for recognition, CoRR, vol. abs/1403.6382, 2014.
- [34] Lowe, David G. (1999). Object recognition from local scale-invariant features. Proceedings of the International Conference on Computer Vision. Vol. 2. pp. 1150–1157. doi:10.1109/ICCV.1999.790410.
- [35] Fukushima, Kunihiro, Upward movement of pattern separation capability of convolutional. Proceedings of the Electronic Information and Communication Society A, 62(10), 650-657, 1979.
- [36] Fukushima, Kunihiro, Location invariance influences the neurocircuit model of the pattern-to-awareness mechanism-neoconnectivity. Proceedings of the Electronic Information and Communication Society A, 62(10), 658-665, 1979.
- [37] Fukushima, K., Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4), 193-202, 1980.
- [38] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 86(11): 2278-2324, 1998.
- [39] Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks[C]. International Conference on Neural Information Processing Systems (NIPS), 1097-1105, 2012.
- [40] Simonyan K, Zisserman A., Very Deep Convolutional Networks for Large-Scale Image Recognition [DB/OL]. [2015-01-02]. <https://arxiv.org/abs/1409.1556>, 2014.
- [41] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-9, 2015.
- [42] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778, 2016.
- [43] Hu J, Shen L, Albanie S, et al. Squeeze-and-Excitation Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(6), 2011-2023, 2017.
- [44] Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 4510-4520, 2018.
- [45] Iandola F N, Han S, Moskewicz M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size [DB/OL]. [2015-01-05]. <https://arxiv.org/abs/1602.07360>, 2016.
- [46] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 580-587, 2014.
- [47] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788, 2016.

- [48] Goodfellow, I., Bengio, Y., Courville, A..Deep learning (Vol. 1).Cambridge: MIT press, 326-366, 2016
- [49] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, L., Wang, G. and Cai, J., Recent advances in convolutional neural networks. arXiv preprint arXiv:1512.07108, 2015.
- [50] Estrach, J.B., Szlam A. and LeCun, Y., Signal recovery from pooling representations. in: Proceedings of the International Conference on Machine Learning (ICML), pp. 307-315, 2014.
- [51] Mathieu, M., Henaff, M. and LeCun, Y., Fast training of convolutional networks through ffts. arXiv preprint arXiv:1312.5851, 2013.
- [52] UFLDL Tutorial - Convolutional Neural Networ...Deep Learning, Computer Science Department, Stanford University [citation needed 2018-11-03]
- [53] Bilinski P, Prisacariu V. Dense Decoder Shortcut Connections for Single-Pass Semantic Segmentation[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6596-6605, 2018.
- [54] Yang M, Yu K, Zhang C, et al. DenseASPP for Semantic Segmentation in Street Scenes[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 3684-3692, 2018.
- [55] Mehta S, Rastegari M, Caspi A, et al. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation[C]. European Conference on Computer Vision (ECCV), 561-880, 2018.
- [56] Pearson, K. (1901) LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2, 559-572.
- [57] Barret. H.H., Foundations of Image Science. John Wiley & Sons, New Jersey, U.K., third edition, 2004.
- [58] Gonzales, R.C. and Woods, R.E., Digital Image Processing. Prentice Hall, second edition, 2002. 795 pages, ISBN 0-201-18075-8.
- [59] Haykin, S., Neural Networks, A Comprehensive Foundations. IEEE Press, USA, 1984.
- [60] Petrou, M. and Bosddogianni, P., Image Processing: The Fundamentals. John Wiley and Sons, Inc., UK, 2000. 335 pages, ISBN 0-471-99883-4.
- [61] Candes, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? Journal of the ACM (JACM), 58(3), 11.
- [62] Abdi. H., & Williams, L.J. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2: 433–459, 2010
- [63] Ahmed, N., Natarajan, T. and Rao, K.R., Discrete Cosine Transform, in IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90-93, Jan. 1974, doi: 10.1109/T-C.1974.223784.
- [64] Li J., Cheng H., Guo H., Qiu S.: Survey on artificial intelligence for vehicles. Automotive Innovation 1(1) 2-14 (2018)
- [65] Li S.: A new vehicle detection method for blind spot detection system based on

DSP. International Journal of Research in Engineering and Science, Vol. 4 Issue 5, 27-29 May (2016)

- [66] Wu B.F., Kao C.C., Li Y.F., and Tsai M.Y.: A real-time embedded blind spot safety assistance system. International Journal of Vehicular Technology Volume 2012, Article ID 506235, (2012)
- [67] Pinard C., Chevalley L., Manzanera A, and Filliat D.: Learning structure-from-motion from motion. arXiv:1809.04471v1 [cs.CV] 12 Sep (2018)
- [68] Wu T.Y., Liu Y.: Position estimation of camera based on unsupervised learning. [arXiv:1805.02020](#) [cs.CV] 5 May (2018)
- [69] Vijayanarasimhan S., Ricco S., Schmid C, Sukthankar. R., Fragkiadaki K.: Learning of Structure and Motion from Video. arXiv:1704.07804v1 [cs.CV] 25 Apr (2017)
- [70] Carvalho M., Saux B.L., Trouvé-Peloux P., Almansa A., Champagnat F.: Deep depth from defocus: how can defocus blur improve 3D estimation using dense neural networks? arXiv:1809.01567v2 [cs.CV] 6 Sep (2018)
- [71] He, K., Zhang, X., Ren, S. and Sun, J., Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv:1406.4729v4 [cs.CV] 23 Apr 2015
- [72] Zhang, X., Zhou,X., Lin, M., and Sun, J., Feature Selective Anchor-Free Module for Single-Shot Object Detection, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 639-647.
- [73] Lindeberg, Tony, Scale Invariant Feature Transform, May 2012Scholarpedia 7(5) DOI: 10.4249/scholarpedia.10491, In book: ScholarpediaChapter: 7(5):10491
- [74] Yohanes, Banu W., Images Similarity based on Bags of SIFT Descriptor and K-Means Clustering, Techné Jurnal Ilmiah Elektroteknika Vol. 18 No. 2, Oktober 2019, Hal 137-146.
- [75] Mostafa Dehghani, Josip Djolonga, etc., Scaling Vision Transformers to 22 Billion Parameters, Computer Vision and Pattern Recognition, arXiv:2302.05442 [cs.CV], Feb. 2023
- [76] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q., Learning Transferable Architectures for Scalable Image Recognition, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, pp. 8697-8710, 2018 doi: 10.1109/CVPR.2018.00907
- [77] He, K., Zhang, X., Ren, S. and Sun, J., Deep Residual Learning for Image Recognition, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778, 2016 doi: 10.1109/CVPR.2016.90
- [78] Road vehicles — Safety of the intended functionality, ISO 21448:2022 <https://www.iso.org/standard/77490.html>

Research achievements

1. Journal Papers

- [1] **Guo, Yuxiang**, Kumazawa, I. and Kaku, C., Application Study of Blind Spot Monitoring System Realized by Monocular Camera with CNN Depth Cues Extraction Approach, SAE International Journal of Connected and Automated Vehicles 2(4):219-232, 2019
【EI Accession Number 20204409405745】

- [2] **Guo, Yuxiang**, Kumazawa, I. and Kaku, C., Blind Spot Obstacle Detection from Monocular Camera Images with Depth Cues Extracted by CNN. Automotive Innovation-Springer. 1, 362–373, 2018
【EI Accession number: 20220411525797】

- [3] **Guo, Yuxiang**, Kumazawa, I. and Kaku, C., Scale variant vehicle object recognition by CNN module of multi-pooling-PCA process. Journal of Intelligent and Connected Vehicles, Accepted 12 Aug 2023
【EI Accession Number: TBA】 (Journal official site IEEE Xplore : <https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=9944931>)

2. Invited Conference presentation

Guo, Yuxiang, Application Study for Blind Spot Obstacle Detection by Deep Learning with Monocular Camera Images, Session of VT2 Intelligent Vehicle Decision and Control. 2018 SAECCCE Congress & Exhibition, Shanghai.

3. Patents

➤ **Invention Patent:**

- [1] CN 110992304 B, [EN] Two dimensional image depth measurement method and its application in vehicle safety monitoring
[**Invention Patent 1/2**], 07.07.2023

➤ **Utility model Patents:**

- [1] CN210554753U, [EN] Electronic parking brake
[**Utility model 1/3**], 2020-05-19 authorized

- [2] CN213008067U, [EN] A detection device of brake light switch
[**Utility model 2/6**], 2021-04-20 authorized

- [3] CN213734954U, [EN] Unexpected power outage protection device and braking system for electromechanical hydraulic electric control braking system,
[**Utility model 3/5**], 2021-07-20 authorized

Acknowledgements

This thesis work is supported by Laboratory for Future Interdisciplinary Research of Science and Technology (Kumazawa Laboratory), Institute of Innovative Research (IIR), (*~April 2016 previous Imaging Science and Engineering Laboratory, Dept. of Information Processing, Interdisciplinary Graduate School of Science and Engineering*) Tokyo Institute of Technology, during the period from October 2017 up to now.

I deeply appreciate Professor Itsuo Kumazawa, my advisor, who gave me many opportunities to be exposed in the research skill building up, besides his normal thesis work advice. Thanks so much to his encouragement leading me to face the challenges and overcome many difficulties.

I thank everyone who had ever helped me during his/her stay in Kumazawa Lab.

Thanks so much to my parents with their persistent and substantial support.

I sincerely acknowledge many of my friends for their supports.

Yuxiang Guo, Aug. 2023@Suzukakedai Campus, Tokyo Institute of Technology