M.S. THESIS

Vanishing Point Detection using Convolutional Neural Network

합성곱 신경망을 통한 소실점 검출

BY

Keunhoi An

February 2020

Department of Computational Science and Technology
Seoul National University

M.S. THESIS

Vanishing Point Detection using Convolutional Neural Network

합성곱 신경망을 통한 소실점 검출

BY

Keunhoi An

February 2020

Department of Computational Science and Technology
Seoul National University

Vanishing Point Detection using Convolutional Neural Network

합성곱 신경망을 통한 소실점 검출

지도교수 강 명 주
이 논문을 이학석사 학위논문으로 제출함

2020년 2월

서울대학교 대학원

계산과학 협동과정

안 근 회

안근회의 이학석사 학위 논문을 인준함

2020년 2월

위 원 장: _____
부위원장: _____
위 원: _____

Abstract

This paper proposes a regression method with one of the standard CNN (convolutional neural network) models, called ResNet (residual neural network), for vanishing point detection. The aim of this thesis is to apply a CNN model for vanishing detection to estimate the position of a vanishing point accurately. Our newly collected KE19 dataset, which used Naver Maps' Street View, is for training the CNN model and comparing it with previous methods. Its main contributions are, (1) applying regression approach to a CNN model, (2) showing improved experimental results compared to the previously proposed work, (3) providing our newly collected KE19 dataset for vanishing point detection, and (4) providing the implications of the effect of combination of L_1 and L_2 loss function to related work. In conclusion we find that the trained CNN model, which is based on ResNet, outperforms previous methods in terms of both computation time and accuracy.

keywords: Vanishing Point Detection, Deep Learning, Convolutional Neural Network, ResNet, KE19 Dataset

student number: 2017-21788

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Deep learning is now one of the major leaders in computer science. There are several factors that enable a big wave of deep learning: (1) big data, (2) developed computation resources, (3) mature cloud service, (4) the trend of the open-source community, (5) the trend of open-publication, and (6) advanced algorithms and models. Among deep learning models, the convolutional neural network (CNN) is essential to computer vision. It can handle high-dimensional feature space much more efficiently than conventional machine learning algorithms, in spite of its inexplicable properties. In the last several years, numerous articles have been devoted to the study of novel CNN architectures, the methodology of training and verification of CNN models and even valuable attempts to explain how CNN models work and the reason why CNN works well.

With the astonishing growth of big data, image and video data have a high portion of real-world data. However, we mainly handle those data as two-dimensional which are not. To restore the spatial information of a real-world scene accurately, we should use specific devices or apply reconstruction algorithms. Vanishing point detection or estimation is the basis of this process.

In a two-dimensional image plane, a vanishing point is the farthest point where mutually parallel lines in three-dimensional real-world space converge after being pro-

jected to a two-dimensional image plane. The vanishing point is also called a direction point, as it has directional information of the original scene in a two-dimensional image plane. Detection of the vanishing point plays an important role in the fields of 3D reconstruction, camera calibration, autonomous driving, etc.

The existing algorithms for vanishing point estimation, which is based on the modeling method, detect the vanishing point by finding the convergence points of mutually parallel lines. As the vanishing point is selected among the intersecting points of the lines, the line segment is the first to be detected in the image. Specifically, by using the Gaussian sphere, the intersection points of the parallel line candidates are the candidates of the vanishing point, and the vanishing point is calculated through a voting scheme. For each line of intersection, all the segments in the image are evaluated, and the point with the largest sum of votes is elected as the initial vanishing point [1]. However, it is difficult to detect the vanishing point of an outdoor image because the image mostly is noisy due to several factors such as device resolution or complex scene structure, which makes it difficult to extract efficient line segments containing strong features.

To solve these problems, several deep learning methods have been proposed for vanishing point estimation recently. The study [2] predicted the vanishing point by using a CNN model. Other approaches are based on VGGNet and AlexNet, and these models treat vanishing point detection as a classification problem or a regression problem [3-5]. These deep learning methods have been proven to work faster and more accurately than modeling-based algorithms [4, 5].

In this paper, we propose a regression method with ResNet for vanishing point detection, which outperforms existing methods. Unlike other deep learning-based vanishing point detection schemes, we use ResNet [6] architecture, which is constructed with residual and identity blocks. Experimental results show that the proposed method has higher accuracy than other vanishing point detection methods: both modeling-based and deep learning-based methods. It is also confirmed that the computational cost is

almost similar to that of the existing deep learning models.

The rest of this paper is organized as follows. In Chapter 2 we present a brief overview of the existing vanishing point estimation methods, and discuss the method of collection of our new dataset, named the KE19 dataset. In Chapter 3 we describe the proposed CNN architecture designed for regression prediction in this study. In Chapter 4 we verify the validity of the proposed model architecture. Finally, in Chapter 5 we conclude our work.

Chapter 2

Related work

2.1 Vanishing point estimation

The intersection of dominant lines appearing in an image in existing modeling-based image processing methods has been considered as the vanishing point [7-9]. A method for identifying the dominant lines is the Gabor filter, which is an orientation-sensitive filter and is used in many areas such as texture segmentation, object recognition, and edge detection. Based on the Gabor filter, H. Kong et al. proposed a vanishing point estimation method using road segmentation and locally adaptive soft-voting [10]. Subsequently, C.-K. Chang et al. obtained a better vanishing point estimation results using Gabor response variance score [11]. T. H. Bui et al. improved the accuracy of vanishing point estimation using lane detection by calculating texture orientation and determining remaining voters [12].

Other methods have also been proposed to estimate the vanishing point through the intersection of dominant straight lines on the image. H.-J. Liu et al. proposed a method of estimation of the vanishing point by deriving the objective function through the calculation of Hough space and finding a value that minimizes this function [13]. Y. Xu et al. proposed a method of estimation of the vanishing point through level-set visualization after calculating the cost functions of extracted straight lines and intersections

[14]. Y. Y. Moon et al. proposed a harmonic search method to estimate the vanishing point by repeatedly performing the Harmony memory method after identifying dominant lines on the image by using Hough transform and Canny edge detection [15].

F. Kluger et al. proposed the first method that applied deep learning to line extraction [16]. In the paper, the vanishing point was estimated using an inverse gnomonic projection of segmented lines.

A. Borji et al. and C.-K. Chang et al. proposed detection methods using deep learning classification without using line segment extraction [3, 4]. In this case, the method in Ref. [3] was derived using VGGNet, and the methods in Refs. [4, 5] were derived using AlexNet. A. Borji et al. obtained an error rate of 5.1% over a 10×10 grid, 15.9% over a 20×20 grid, and 25.3% over a 30×30 grid through the Top-5 grid image [3]. C.-K. Chang et al. proposed an improved method that showed an accuracy of 92.09% within the error range of [5°, 5°] at 225 (15×15 grid) class using AlexNet. In particular, C.-K. Chang et al. proposed a method of collecting vanishing points data through a novelty method [4]. Y. Shuai et al. proposed a method for detecting the vanishing point through a regression approach using AlexNet [15]. This method also exploits vanishing points using AlexNet, but there is a disadvantage in that the geometric information can be lost while passing through the last two fully connected layers before the output layer.

2.2 Dataset collection and augmentation

In ICRA, C.-K. Chang et al. constructed a vanishing point dataset using Google Street View [4]. This has considerably improved the number of training datasets of existing vanishing points. Moreover, the dataset has following merits: (1) it covers a wide range of road appearance; (2) it has a panorama capability which can generate multiple VP views from individual locations; (3) it provides camera parameters such as pitch information to the estimate horizon line.; and (4) we can easily augment image data by

changing the angle of viewpoint with Google Maps. However, the dataset proposed in the paper has following disadvantages.

The vanishing point used in the lane detection is set to the point where the intersecting points of the dominant segment lines are extracted from the tangent lines on the image. In the paper, the authors increase the number of data from a single image by changing the angle of view. However, the vanishing points of the newly extracted image and the original image differ from each other as shown in Figure 2.1. (New Zealand: 9209, 9261, and 9317 respectively)

The reason for this difference is as follows. The AVA image data [17] '282451.jpg' is shown in Figure 2.2. The location of the ground truth is well represented, which is indicated by the red star in the original image on the curve road (left). However, when cropping the original image by changing the focus direction of the camera the position of the vanishing point, which is the intersection point, also changes as indicated by the sky blue star. As shown in Figure 2.2 (right), the position of the vanishing point disagrees with its accurate position.

This problem also occurs when there is a slight curvature in the image on straight roads. For instance, from '290184.jpg' of the AVA data shown in Figure 2.3, we can observe that the position of the ground truth is indicated by the red star in the original image (left). However, when the method similar to the single-shot image collection proposed in the aforementioned paper is performed, we observed that the newly annotated intersection point (sky blue star) of the dominant tangent line indicated in yellow is different from the actual ground truth (right).

With those inaccurate datasets, the model can only occur underfitting, which is the state that has not reached the proper decision boundary, even the number of its parameters is large. Therefore, we built a crop-based single image dataset of straight road view images. We used Naver Maps' Street View to extract the image of a straight road on an expressway in the Republic of Korea (South Korea). Dataset-1 was collected from the starting point of Incheon International Airport Expressway to Unseo Station. Dataset-2

was collected from Seocho interchange to Suwon Singal interchange of Gyeongbu Expressway. Dataset-3 was collected from Sannae interchange to Deogyusan Rest Area of Tongyeong Daejeon Expressway. Each sub-dataset consists of 100 images.

Most of the common vanishing points detected in autonomous driving are located at the center of the image. When a neural network model is only trained with such data, most of the estimated coordinates are very likely to be distributed in the central portion of the image. Therefore, to prevent this phenomenon, in this paper our data was augmented similarly to the method used in DeepVP [4]. Through this augmentation, the position of the vanishing point was evenly distributed in each image.

We cropped the collected image as follows. In the image obtained from the Naver Maps' Street View, as shown in Figure 2.4, the number of the grid is set as 15×15 around the vanishing point. Subsequently, the image of size 340×340 was cropped through the sliding window approach method. Thus, the vanishing point can be evenly distributed on the cropped images. Simultaneously, the vanishing point is prevented from being present only on the grid by moving the sliding window randomly within [0,15] pixels. We excluded the cropped image if the vanishing point is not located in the image.

Through the above-mentioned method, each original sub-dataset has been increased to 22,479, 22,500, and 22,500. As shown in Table 2.1, most of the existing vanishing point datasets have less than 30,000 images, except for DeepVP-1M. Our KE19 dataset will be released later on http://ncia.snu.ac.kr/xe/.

Year	Dataset	Scene Class	Reference	Total image
2003	York Urban Lane Segment	2	[18]	102
2009	Kong'09	3	[10]	1,003
2011	Eurasian Cities	1	[19]	103
2012	PKU Campus	1	[20]	200
2012	Chang'12	2	[11]	25,076
2014	Le'14	1	[21]	16,000
2015	Tvanishing pointD	2	[22]	102
2017	Zihan'17	2	[9]	2,275
2018	DeepVP-1M	23	[4]	1,053,425
2019	**KE19**	3	-	67,479

Table 2.1: Overview of Most Available Vanishing Point Dataset and KE19 Dataset

Figure 2.1: Example of the false ground truth label of DeepVP-1M: New Zealand

Figure 2.2: Example of the false crop in curve road of AVA dataset

Figure 2.3: Example of the false crop in the straight road of AVA dataset

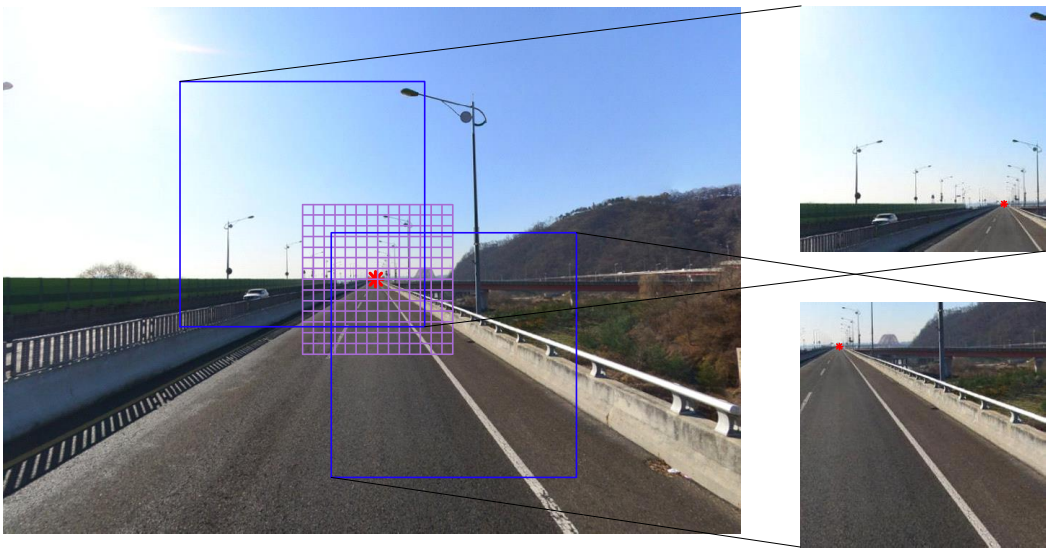

Figure 2.4: Single shot image crop: left - original Naver Maps' Street View. right - crop image near vanishing point

Chapter 3

Model architecture

3.1 ResNet architecture

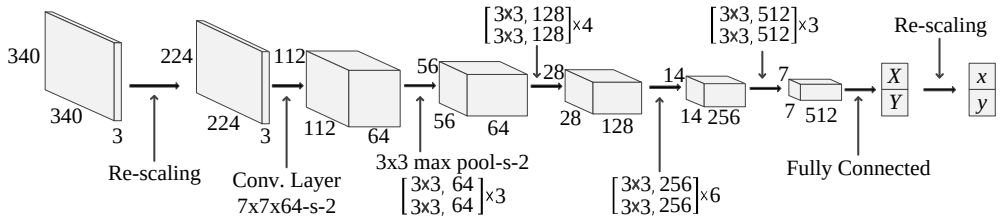

Figure 3.1: Architecture of the proposed Network - Regression ResNet-34

In this study, ResNet [6] was used to detect the vanishing point because of the following reasons. First, if pooling is used repeatedly in the architecture, the characteristics of the vanishing point may be weakened. To prevent this, ResNet has the advantage of using less pooling compared with other networks. Second, most neural networks have fully connected layers before the output layer, and thus, they have the disadvantage of losing geometric information of input data as it passes through these layers. ResNet has the advantage of preserving geometric information of input data because it does not have fully connected layers, except for the output layer. Third, ResNet has various depths and each depth has its pros and cons in the aspect of computational cost and

accuracy. Hence it is one of the best benchmark CNN models to be applied for transfer learning, which adopts a pre-trained model to the task or gives a slight change to the model for a new task.

The architecture of ResNet-34 used in the study is shown in Table 3.1. The size of the input image is set to 340 $\times$ 340, but the original input size of ResNet is set to 224 $\times$ 224. Therefore we first resized an input image and added it to the network. We set the output layer as a 2 $\times$ 1 vector layer to predict (X, Y) coordinates directly, which is different from other vanishing point detection works [3,4,16]. Thereafter, the result obtained for the image of 224 $\times$ 224 was re-scaled again to 340 $\times$ 340 to obtain the final predicted coordinate value (x, y). A description of the overall structure is shown in Figure 3.1.

3.2 Combined L_1 - L_2 loss function

The designed loss function for the architecture is as follows. The distance between the estimated vanishing point(x_p, y_p) and the ground truth vanishing point(x_t, y_t) is calculated using the L_2 norm, so it can be the one term of the loss function. Then, we add L_1 norm with a hyper-parameter λ. The reason we combine L_1 norm with L_2 norm is that L_1 norm is larger than L_2 norm if the predicted location of vanishing point is far from the ground truth since L_1 distance is also named as Manhattan distance. This makes it is possible to further reduce the error between (x_p, y_p) and (x_t, y_t). Therefore, the loss function used in this paper is proposed as follows.

$$Loss = L_2\left(x_t - x_p, y_t - y_p\right) + \lambda L_1\left(x_t - x_p, y_t - y_p\right)$$

Through numerous experiments, we have observed that setting $\lambda = 0.3 \sim 0.7$ yields the best results, while pure L_1 or L_2 loss function cannot get closer ground truth position within a certain radius as shown in Figure 3.2. Hence, in this study, training and evaluation are performed by setting $\lambda = 0.5$.

However, $\lambda < 1$ is collide with our motivation. The main purpose of the combined

loss function is to let predicted vanishing point approach to ground truth position much faster since L_1 norm has a higher value than L_2 norm. We've tried combined loss function with $\lambda > 1$ but the accuracy and the converge speed were worse than pure L_2 norm. Qualitative analysis for this result is that the problem of space searching is greatly affected by the 'scale' problem. As we train our model, learning scheduling is one of the main methods to obtain state-of-the-art performance. Without this method, we've found that neither AlexNet based model or ResNet based model cannot reach high accuracy performance. The combined loss function with $\lambda > 1$ is not only mainly change its converge direction with the derivative of L_1 norm, but also very likely to 'move' a bigger step than pure L_2 norm or combined loss function with $\lambda < 1$ in each iteration. Since the learning scheduling is essential to our experiment, which implies that scale problem is hugely important, the combined loss function with $\lambda < 1$ is better than $\lambda > 1$ for searching better local optimum buried in small scale rugged valleys.

Layer name	Output size	34-layer
Input	224×224	-
Conv1	112×112	7×7, 64, stride 2
Conv2_x	56×56	3×3, max pooling, stride 2 $\begin{bmatrix} 3\times3, & 64 \\ 3\times3, & 64 \end{bmatrix} \times 3$
Conv3_x	28×28	$\begin{bmatrix} 3\times3, & 128 \\ 3\times3, & 128 \end{bmatrix} \times 4$
Conv4_x	14×14	$\begin{bmatrix} 3\times3, & 256 \\ 3\times3, & 256 \end{bmatrix} \times 6$
Conv5_x	7×7	$\begin{bmatrix} 3\times3, & 512 \\ 3\times3, & 512 \end{bmatrix} \times 3$
Output	1×1	Average pooling 2-d fc, softmax

Table 3.1: Proposed model based on ResNet-34 Architecture

Hardware	**Specification**
CPU	Inter Core i5-6500
GPU	Tesla K80
GPU Memory capacity	12 GB
Software	**Specification**
Python	Version: 2.7.12
TensorFlow	Version: 1.12.0
Operation System	Linux Ubuntu 16.04

Table 3.2: Data Sheet of Training Environment

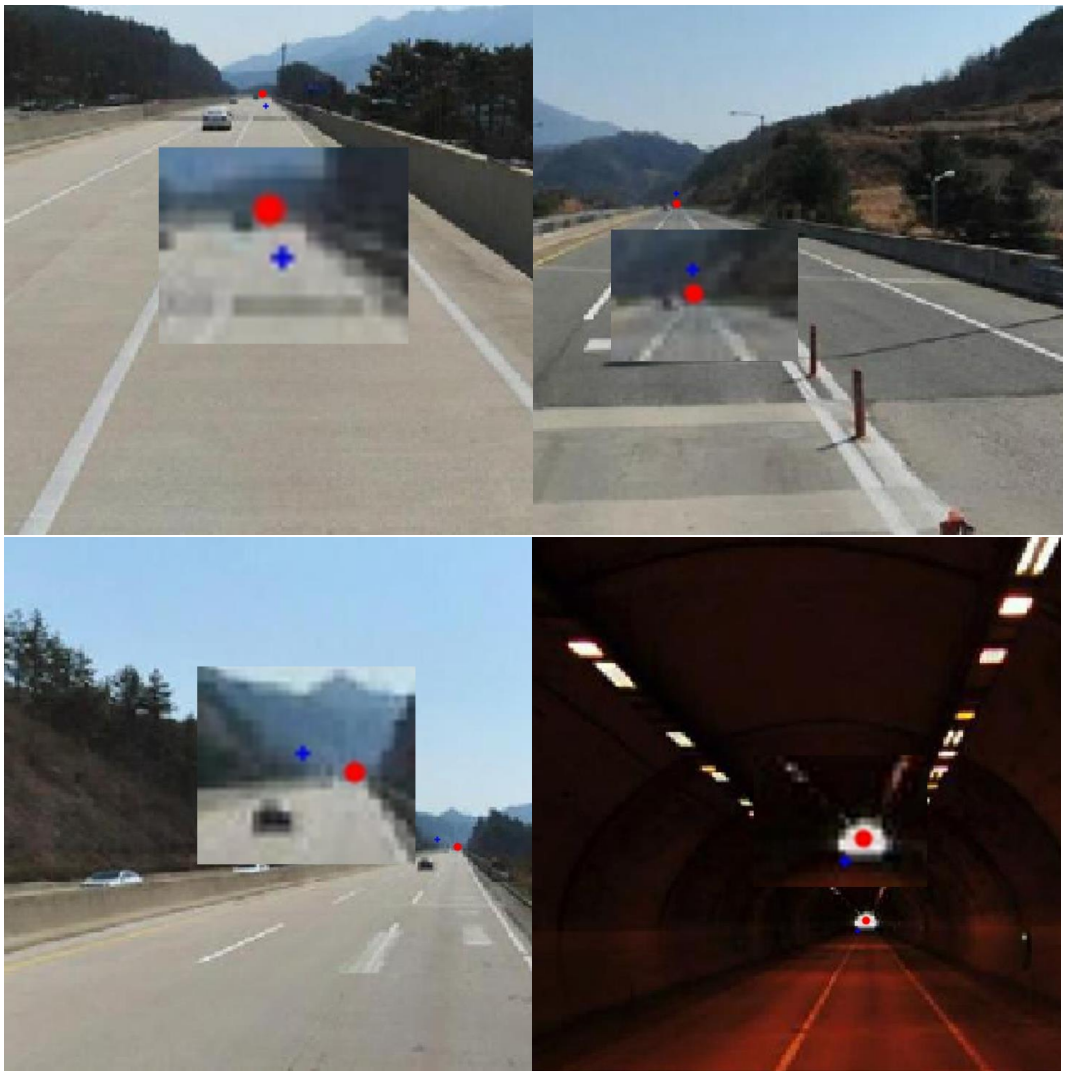

Figure 3.2: Incomplete converge result with pure L_2 norm loss function

Chapter 4

Experimental result

4.1 Comparison with other methods

In this section we verify the validity of the proposed architecture through our experiment. The experiment was conducted in the environment described in Table 3.2. The epoch was set to 70, and hence, the total number of iterations was approximately 29,500. Through the learning rate scheduling, we set the learning rate as η = 10^{-4} by iterations 1000, as 10^{-5} by iterations 10,000, as 5×10^{-6} by iterations 15,000, and as 106 beyond. The reason for applying such learning rate scheduling is from intuition. As the training process goes on, a better local minimum is near in much sharper saddle points. To approach those minimum points, the learning rate needs to be reduced. The batch size was set to 128, and the optimizer used was Adam optimizer.

To train and test our model and other methods, 80% of each sub-dataset was used as training data, and 20% was used as test data. In the training process, we shuffled the training data in each epoch to minimize the gap between the training data and test data.

The algorithms used for the comparison are the Gabor texture vanishing point [12] and Reg Conv [5]. The result is shown in Table 4.1. As shown in Table 4.1, our proposed method achieves the best performance in datasets 1, 2, and 3.

	Gabor Texture vanishing point	Reg Conv	**Reg ResNet-34**
Dataset-1	6.282533	11.740226	**4.135742**
Dataset-2	11.002503	16.352880	**6.102345**
Dataset-3	3.707873	9.283470	**3.528673**
Average	6.997636	12.459071	**4.589054**

Table 4.1: The result of the experimental result: $Loss$ error

The experimental results are shown in Figure 4.1–3. The red circle represents the ground truth, the green x represents the result of Reg Conv, the yellow triangle represents the result of the Gabor texture vanishing point, and the blue cross represents the result of Regression ResNet-34.

The time cost per image was 56.8246 s for the Gabor texture vanishing point and approximately 3.916×10^{-3} s for both Reg Conv and Regression ResNet-34.

In the case of the Gabor texture method, we observe that detecting the vanishing point by extracting the tangent line through road detection is effective. However, when there are distinct straight lines in the image area other than the driving road area, it is confirmed that the vanishing point deviates greatly from the ground truth. For example, as shown in Figure 4.4, the vanishing point detection may be inaccurate if the image shows strong lines in the brick pattern area, shadow boundary, and overpass boundary. In Figure 4.4, the red line segment is the predicted road detected by the algorithm, and the point marked with the blue rectangle is the predicted vanishing point.

Moreover, the Gabor texture vanishing point has a disadvantage in that the processing requires a large time cost per image. In contrast, our proposed method has an advantage in that the vanishing point can be detected most accurately even at a higher speed.

Figure 4.1: Examples of dataset-1. The red circle is Ground truth, green x-mark is Reg Conv, the yellow triangle is the Gabor Texture vanishing point and the blue cross is Reg ResNet-34.

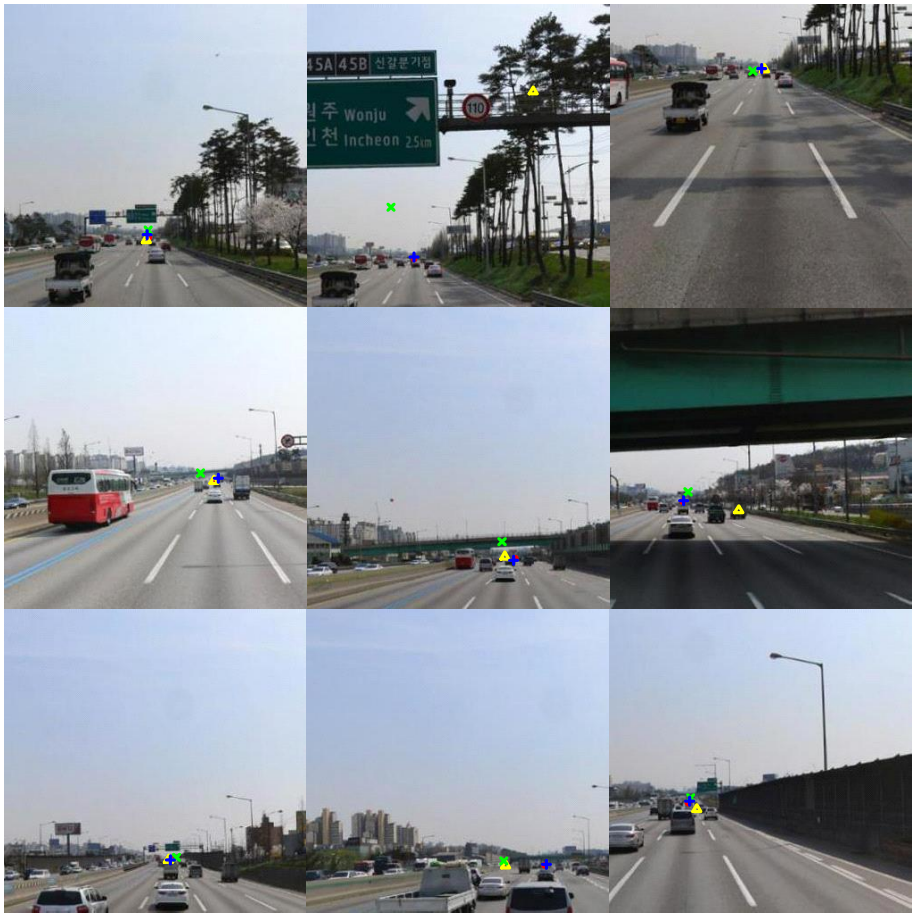

Figure 4.2: Examples of dataset-2. The red circle is Ground truth, green x-mark is Reg Conv, the yellow triangle is the Gabor Texture vanishing point and the blue cross is Reg ResNet-34.

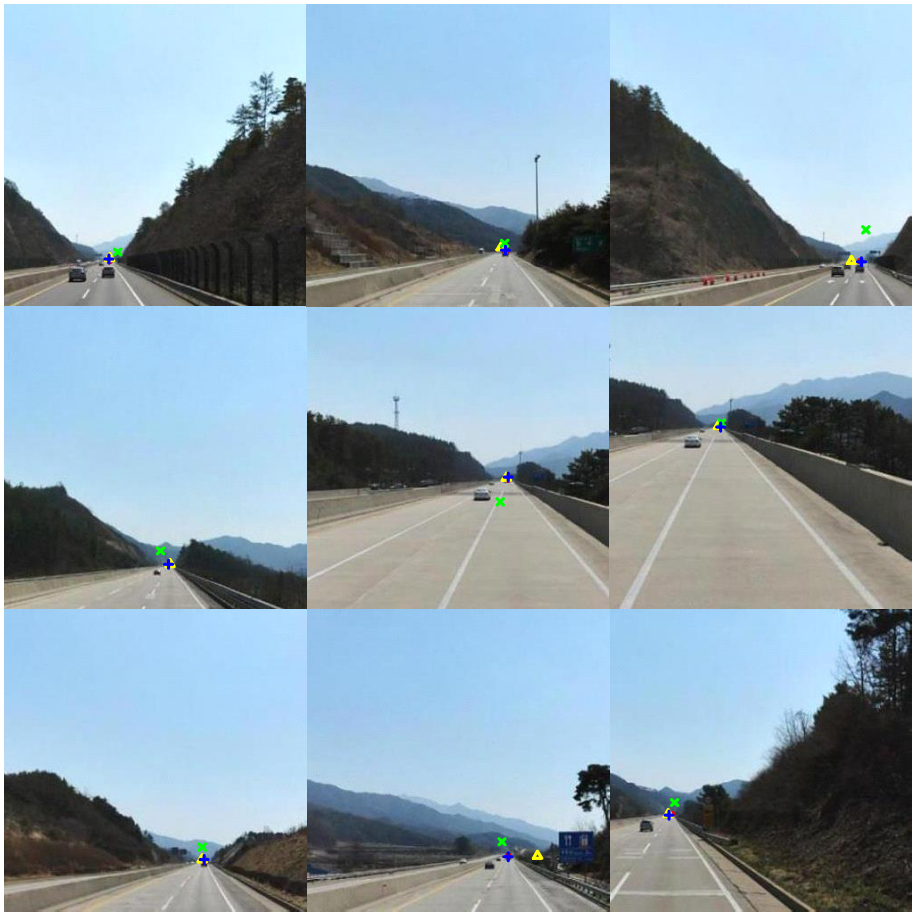

Figure 4.3: Examples of dataset-3. The red circle is Ground truth, the green x-mark is Reg Conv, the yellow triangle is the Gabor Texture vanishing point and the blue cross is Reg ResNet-34t.

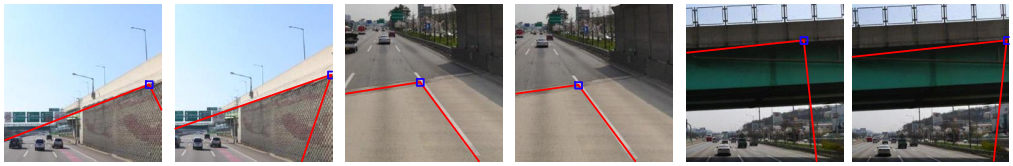

Figure 4.4: Bad result of Example - The Gabor Texture vanishing point

The experiment does not include other benchmark datasets since there are some errors in the previous vanishing point dataset, as we described in Figure 2.1-4.

4.2 Comparison: architecture of model depth

In this study, we performed experiments on all five architectures mentioned in the residual neural network paper. The architecture that yields the best results is ResNet-101 as shown in Figure 4.5. However, as the environments for performing vanishing point detection are mostly limited systems, such as autonomous driving, ResNet-34 is considered to be the most suitable model for real-life applications, considering the operational limits of a mobile device system Table 4.2.

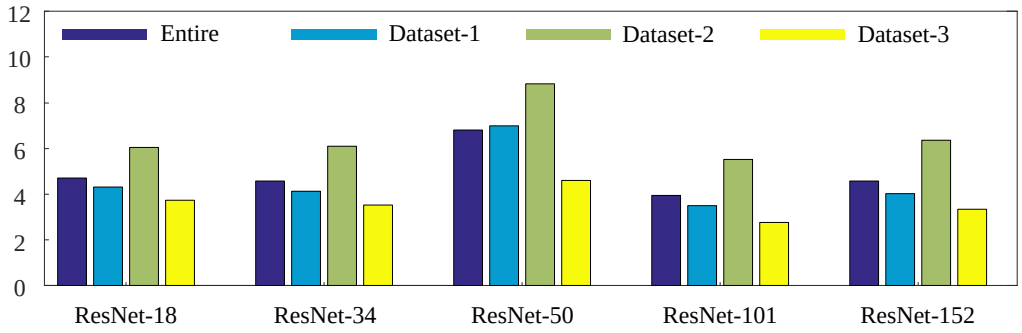

Figure 4.5: Comparison Result of Residual Network - 18, 34, 50, 101 and 152: L_2 error

	Entire	Dataset-1	Dataset-2	Dataset-3
ResNet-18	4.705476	4.317501	6.057426	3.741157
ResNet-34	4.589054	4.135742	6.102345	3.528673
ResNet-50	6.806540	6.981094	8.833066	4.605615
ResNet-101	**3.934230**	**3.498600**	**5.533292**	**2.770410**
ResNet-152	4.579554	4.031265	6.353890	3.353020

Table 4.2: Comparing the results of Residual Network - 18, 34, 50, 101 and 152: $Loss$ error

Chapter 5

Conclusion

This paper proposes a regression method with a residual neural network for vanishing point detection. Also, a method of collection of a dataset for training the network is described and a newly collected public dataset named KE19 dataset is provided for vanishing point detection. It is shown that the proposed method obtains the most accurate result at a faster speed than the other methods that are widely used for vanishing point detection. Experiments on model depth have also shown that ResNet-34 can be more efficient than other ResNet architectures.

We hope that this result will be applied for camera calibration, reconstruction of 3D space, and autonomous driving.

Bibliography

[1] Zongsheng Wu, Weiping Fu, Ru Xue and Wen Wang. “A Novel Line Space Voting Method for Vanishing-Point Detection of General Road Images”. *Sensors 2016, 16, 948*, 2016

[2] R. Itu, D.Borza and R. Danescu. “Automatic extrinsic camera parameters calibration using Convolutional Neural Networks”. *2017 IEEE 13th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2017

[3] Ali Borji. “Vanishing Point Detection with Convolutional Neural Networks”. 2016. Available: https://arxiv.org/abs/1609.00967

[4] Chin-Kai Chang, Jiaping Zhao and Laurent Itti. “DeepVP: Deep Learning for Vanishing Point Detection on 1 Million Street View Images”. *IEEE International Conference on Robotics and Automation (ICRA)*, 2018

[5] Yan Shuai, Yan Tiantian, Yang Guodong and Liang Zize. “Regression Convolutional Network for Vanishing Point Detection”. *2017 32nd Youth Academic Annual Conference on Chinese Association of Automation (YAC)*, 2017

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. “Deep Residual Learning for Image Recognition”. *Computer Vision and Pattern Recognition (CVPR)*, 2016

[7] Company P., Varley P.A.C., Plumed R. “An Algorithm for Grouping Lines which Converge to Vanishing Points in Perspective Sketches of Polyhedra”. *Graph-*

ics Recognition. Current Trends and Challenges. GREC 2013. Lecture Notes in Computer Science, vol 8746. Springer, Berlin, Heidelberg, 2014

[8] Pedro Miraldo, Francisco Eiras and Srikumar Ramalingam. "Analytical Modeling of Vanishing Points and Curves in Catadioptric Cameras". *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2012-2021*, 2018

[9] Zihan Zhou, Farshid Farhat and James Z. Wang. "Detecting Dominant Vanishing Points in Natural Scenes with Application to Composition-Sensitive Image Retrieval". *IEEE Transactions on Multimedia, Vol 19, Issue 12*, 2017

[10] Hui Kong, Jean-Yves Audibert and Jean Ponce. "General Road Detection From a Single Image". *IEEE Transactions on Image Processing, Vol. 19, No.8, pp.2211-2220*, 2010

[11] Chin-Kai Chan, Christian Siagian, Laurent Itti. "Mobile Robot Monocular Vision Navigation Based on Road Region and Boundary Estimation". *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1043-1050*, 2012

[12] Trung Hieu Bui, Takeshi Saitoh and Eitaku Nobuyama. "Road Area Detection based on Texture Orientations Estimation and Vanishing Point Detection". *The SICE Annual Conference 2013*, 2013

[13] Liu Hua-jun, Guo Zhi-bo, Lu Jian-feng and Yang Jing-yu. "A Fast Method for Vanishing Point Estimation and Tracking and Its Application in Road Images". *2006 6th International Conference on ITS Telecommunications*, 2006

[14] Yiliang Xu, Sangmin Oh and Anthony Hoogs. "A Minimum Error Vanishing Point Detection Approach for Uncalibrated Monocular Images of Man-Made Environment". *2013 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1376-1383*, 2013

[15] Yoon Young Moon, Zong Woo Geem and Gi-Tae Han. “Vanishing point detection for self-driving car using harmony search algorithm”. *Swarm and Evolutionary Computation, Vol 41, pp.111-119*, 2018

[16] Florian Kluger, Hanno Ackermann, Michael Ying Yang and Bodo Rosenhahn. “Deep Learning for Vanishing Point Detection Using an Inverse Gnomonic Projection”. *39th German Conference on Pattern Recognition (GCPR), pp.17-28*, 2017

[17] Naila Murray, Luca Marchesotti, Florent Perronnin. “AVA: A Large-Scale Database for Aesthetic Visual Analysis”. *2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2408-2415*, 2012

[18] P. Denis, J.H. Elder and F.J. Estrada. “Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery”. *Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part 2, pp. 197-210*, 2008

[19] Tretyak, E., Barinova, O., Kohli, P. et al. “Geometric Image Parsing in Man-Made Environment”. *Int J Comput Vis (2012) 97: 305. https://doi.org/10.1007/s11263-011-0488-1*, 2010

[20] Bo Li, Kun Peng, Xianghua Ying and Hongbin Zha. “Vanishing point detection using cascaded 1d Hough transform from single images”. *Pattern Recognition Letters, Vol 33, Issue 1, 1 January 2012, Pages 1-8*, 2012

[21] Manh Cuong Le, Son Lam Phung, Abdesselam Bouzerdoum. “Lane Detection in Unstructured Environments for Autonomous Navigation Systems”. *ACCV 2014: Asian Conference on Computer Vision, Springer, pp. 414-429*, 2014

[22] Angladon, Vincent and Gasparini, Simone and Charvillat, Vincent. “The Toulouse Vanishing Point Dataset”. *Proceedings of the 6th ACM Multimedia Systems Conference (MMSys ’15)*, Portland, OR, United States, 2015

초 록

본 논문은 합성곱 신경망을 통하여 소실점의 위치를 정확하게 추정하는 것을 목표로 한다. 본 논문에서 저자는 회귀분석을 통한 소실점 검출을 수행하는 ResNet 기반의 합성곱 신경망을 제안한다. 소실점 검출을 위한 공개 데이터셋이 부족한 현 상황을 극복하고자 본 논문에서는 네이버 지도의 로드뷰 데이터를 활용하여 새롭게 수집된 KE19 데이터셋을 통하여 제안 모델의 학습 및 이전의 모델과의 성능 비교를 진행하였다. 본 논문은 (1) 회귀분석 방법론을 합성곱 신경망에 적용, (2) 이전 모델보다 발전된 실험적 결과, (3) 새롭게 수집된 KE19 데이터셋을 공공 데이터셋으로 제공, 그리고 (4) 관련 분야에서 L1-L2 손실 함수의 결합이 가지는 효과를 암시하는 것에 대한 기여도를 가진다고 판단된다. 실험 결과로 알 수 있듯이 ResNet 기반으로 훈련된 제안 모델은 시간 비용과 정확도 방면 모두에서 이전 모델들보다 성능이 뛰어남을 보였다.

주요어: 소실점 검출, 딥러닝, 합성곱 신경망, ResNet, KE19 데이터셋

학번: 2017-21788

ACKNOWLEDGEMENT

졸업을 앞둔 시기, 새벽에 앉아 차분히 회상해보니 정말 많은 분들의 도움이 있었기에 석사 과정을 마무리할 수 있었음을 다시금 깨닫게 됩니다.

먼저 다양한 기업 프로젝트 경험와 공부하기 좋은 환경을 제공해주신 강명주 지도교수님에게 깊이 감사드립니다. 입학부터 졸업까지, 대학원 생활동안 지도교수님의 지원과 지지가 있었기에 서울대학교 대학원 생활을 보낼 수 있었습니다. 또한 바쁘신 와중에도 흔쾌히 본 학위논문 심사를 맡아주신 김종암 교수님과 정미연 교수님께도 큰 감사의 말씀을 드립니다.

연구실 생활을 하면서 훌륭한 분들과 함께 일하고 시간을 보내며 추억을 쌓을 수 있어서 정말로 행복했습니다. 먼저 학기 초부터 저를 지도하고 전심으로 아껴주고 챙겨주던 한수 형에게 감사드립니다. 항상 성실한 모습으로 목표하신 일들 모두 형통하게 이루시길 기도하겠습니다. 그리고 동기로 같이 들어온 서현이 형에게도 감사드립니다. 표현하진 않았지만 형이 저의 동기라는 것이 제게는 큰 행운이였습니다. 또한 2년 동안 같은 방 식구로 함께 하셨던 김종태 책임님에게도 감사드립니다. 함께 하는 시간동안 많이 감사했습니다. 이 외에도 이 글을 읽으시는 모든 분들에게 감사의 말씀을 전해드립니다. 계산과학 선후배들과 수리과학부 분들 모두 형통한 대학원 생활 보내시길 기도하겠습니다.

돌이켜보면 기쁜 일도, 힘든 일도 많았던 대학원 생활이었습니다. 한결같은 마음으로 항상 응원하고 사랑을 표현해준 가족이 없었다면 견디기 힘든 시간이 되었을 것 같습니다. 부족한 아들을 항상 자신보다 사랑해주는 어머니와 아버지, 그리고 멀리 캐나다에 있는 항상 보고싶은 동생 민아에게 누구보다 사랑하고 고맙다는 말을 전하고 싶습니다.

마지막으로 제 삶을 놓지 않으시는 하늘에 계신 분에게도 감사드립니다.