Two Approaches for Recognizing Roadworks and Crosswalks in Traffic Scenes

Linchen Zhu
Department of Informatics
Karlsruhe Institute of Technology
Email: uaiup@kit.edu

Mushi Duan
Department of Informatics
Karlsruhe Institute of Technology
Email: uieer@kit.edu

Yuting Wang
Department of Informatics
Karlsruhe Institute of Technology
Email: utdyc@kit.edu

***Abstract*—This paper presents a comparison between two approaches for recognizing roadworks and crosswalks in traffic scenes. In recent years, convolutional neural networks (CNN) have been successfully employed in a range of tasks in computer vision; we also utilized CNNs for scene recognition in this practical course. One of the approaches we have implemented is a combination of the CNN and the support vector machine (SVM), where, in this approach, the CNN is utilized as a feature extractor. The other approach is where a pre-trained CNN is fine-tuned on a more specific dataset. It should be noted that the experimental results suggest the second approach can achieve a significantly improved performance. In order to explain this improvement, we have implemented feature visualization using the Deconvolutional Network (deconvnet). By employing this, we can intuitively compare the features learned by the fine-tuned and the original CNN.**

1. Introduction

For both autonomous cars and computer assisted driving, sometimes it's very helpful when the environment surrounding the car can be recognized. For example, when the car drives through roadworks, the driver should drive more carefully. Or, when a crosswalk is encountered, the driver should slow down. In Germany, roadworks are normally signposted with a continuous line of road signs in red and white, while crosswalks are marked with white and black stripes. Owing to this signage, roadworks and crosswalks can be distinguished from other traffic scenes very easily. In this practical course, we aim to recognize roadworks and crosswalks in traffic scenes, and it should be noted that our implementation can be extended to other similar scene recognition tasks without effort.

In the field of computer vision, scene recognition has always attracted the attention of researchers; numerous algorithms for scene recognition have been proposed. For example [1] learns high-level representations of the scene categories based on local patches, and [2] learns global features of the scene categories through energy spectra and spectrogram. In recent years, due to the growing computational power of GPUs, which makes the effective training of CNN with deeper architecture on larger datasets possible, CNNs have shown excellent performance in various visual recognition tasks; as a result, they have become state-of-the-art in object recognition tasks [4] and scene recognition tasks [3]. One of the most important reasons why CNNs have achieved these impressive improvements is that they can learn the discriminative and progressively abstract features of images. Therefore, in order to investigate the outstanding feature learning ability of CNNs, we employ the CNN-based approaches in this practical course for the scene recognition tasks described above.

The layout of this paper is as follow. Section 2 first provides an overview of the related work, after that some background knowledge of the CNN and an introduction of the deep learning framework Caffe is given in Section 3. Section 4 introduces the two implemented approaches in detail respectively. The performance of the two approaches was evaluated and compared through experiments, and the experimental results are demonstrated in Section 5. Section 6 discusses the most relevant conclusions of our work.

2. Related Work

Donahue et al. [5] have investigated whether the deep features learned by the CNN trained on object recognition tasks, such as ImageNet, can be repurposed to new tasks like scene recognition. Their work has proven that a simple, linear classifier (e.g. SVM or LogReg), which is based on the features learned by CNN, outperforms the state-of-the-art hand-engineered features, with a recognition accuracy improvement of 2.9% in the scene recognition task. This is despite the fact that their CNN is trained on ILSVRC, an object recognition database. This shows the efficiency of the feature learning ability of CNN, and, additionally, inspires the idea behind our work.

In most previous work, CNNs were trained primarily on object-centric datasets, like ImageNet. However, the features learned by object recognition CNNs sometimes may not perform competently for scene recognition tasks. In [3] a new scene-centric dataset called Places has been introduced for training CNNs. As mentioned in [3], the scene-centric dataset contains images with richer and more diverse visual information compared with the object-centric dataset. The Places dataset has 476 scene categories with a

total of 7 million images; as such, it is the largest scene-centric dataset up to present. Experiments show that CNNs trained on the Places dataset have achieved significantly higher recognition accuracy in scene recognition tasks compared with previous CNNs trained on ImageNet.

Girshick et al. [6] have introduced the CNN fine-tuning approach. After the CNN is pre-trained on the ILSVRC 2012 dataset, the number of nodes of the output softmax layer is modified to the number of classes of the new problem domain. Following this, the CNN is further trained on region proposals from PASCAL VOC. Due to the outstanding performance of their approach on the object detection task, they believe that the "supervised pre-training/domain-specific fine-tuning" paradigm will be one of the most efficient ways in which to handle sparse training data problems. Oquab et al. [7] have proposed a similar approach for reusing the parameters learned on ImageNet dataset; this will enable the computation of mid-level features for images in another dataset with limited training data. They first trained a CNN on the source task, followed by the softmax layer FC8 of the pre-trained CNN being replaced by a fully connected layer FCa and, in addition, a softmax layer FCb. Subsequently, the new CNN was trained on the labeled images of the target task, with pre-trained layers being frozen.

3. CNN and Caffe Framework

A Convolutional Neural Network is a computing model motivated by the human brain. Like the human brain, it is composed of a large number of elementary units called neurons. Furthermore, neurons in the CNN are arranged in layers, the complexity and the capacity of the CNN depends on the number and the size of the layers. Different to ordinary neural networks, two assumptions have been made by the CNN on the property of its input images: the "stationarity of statistics and locality of pixel dependencies" [4]. With the help these two assumptions, CNNs can be much less complex than ordinary neural networks, while the capacity of CNNs is only slightly affected.

In 2012 Alex Krizhevsky et al. proposed a novel CNN architecture, which was later called Alex-Net [4]. The CNN with their proposed architecture has significantly outperformed the second best results in the ImageNet LSVRC 2010 competition, since then the CNN approach became more and more popular on various visual recognition tasks. Some highlights of Alex-Next include ReLU nonlinearity, training on multiple GPUs, local response normalization, and overlapping pooling. Replacing the conventional saturating nonlinearities like tanh with the non-saturating nonlinearity ReLU can notably reduce the training time. Spreading the CNN across multiple GPUs facilitates the training of a larger network. Local response normalization can be regarded as a brightness normalization scheme, which effectively reduces the test error rate. The last novel feature, overlapping pooling, can help to prevent overfitting.

Figure 1 illustrates the architecture of Alex-Net. It is composed of eight layers, among which the first five layers are convolutional layers and the last three are fully-connected layers. The entirety of Alex-Net is split in two columns since it is trained on two separate GPUs, except the 3rd convolutional layer the other convolutional layers in two columns are independent. AlexNet takes a 224*224*3 image as input, and then filters the image with the 1st convolutional layer which consists of 96 kernels of size 11*11*3, followed by ReLU nonlinearity, response-normalization layers, and the max-pooling layers. Detailed information relating to other hidden layers is shown in figure 1. The output layer of Alex-Net is a 1000-way softmax layer, which produces the a posteriori probabilities of the 1000 image classes, given the input image.

Besides Alex-Net, there are also many other CNN architectures been proposed, for example LeNet [9], GoogLeNet [11] and VGGNet [10]. Among them Alex-Net is one of the most extensively used CNN architectures at the present moment in the community; much previous work has been conducted based on Alex-Net, such as [6] [14] [3]. In these studies, some small modifications have been applied to Alex-Net, which include different training dataset, different sizes of input layer, and other related modifications. In this work we also employed Alex-Net architecture, owing to the fact that many pre-trained models with very similar architecture can be found in the Caffe Model Zoo, including Places-CNN, which is used throughout this work.

In this work, all the experiments relating to the CNN, including feature extraction, feature visualization, CNN training, and testing, were performed using Caffe [12]. Caffe is a deep learning framework developed by the Berkeley Vision and Learning Center. Some highlights of Caffe include modularity, separation of representation and implementation, test coverage, Python & MATLAB bindings, and pre-trained reference models. Two unique features of Caffe are worth mentioning. First of all, Caffe is implemented in C++. This not only ensures the high efficiency of the programs execution, but also makes Caffe easy to integrate into the existing C++ environment. In addition, Caffe supports both GPU and CPU modes. The other unique feature of Caffe worth mentioning is that a variety of pre-trained Caffe models are free to access at the so-called Caffe Model Zoo; this allows a quick start of experiments based on state-of-the-art results.

4. Two Approaches to Train Scene Classifiers

In this section, the two approaches we have implemented are discussed in detail. Our first approach is similar to the work presented in [5], which utilizes a pre-trained CNN to extract deep representations of images. Following this, a SVM classifier is trained on those representations of images. In our second approach, the CNN is not just used as a feature extractor, but is also used as a classifier, which means that

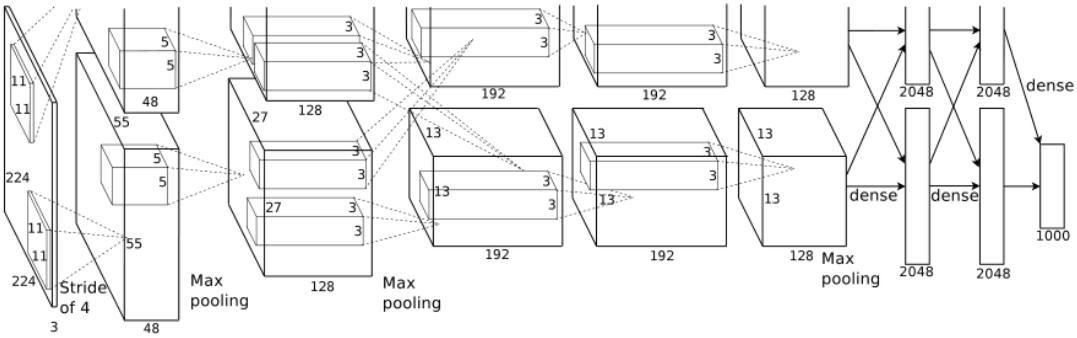

Figure 1: *Alex-Net architecture: (conv → relu → norm → pool) * 2 → (conv → relu) * 2 → (conv → relu → pool) → (fc → relu → drop) * 2 → fc. Image copied from [4]*

the CNN can directly predict whether an image represents roadworks or not. More importantly, the parameters of the CNN learned from the source task are further fine-tuned on the domain-specific training data.

4.1. CNN features + SVM

As shown in [3], the CNN trained on scene-centric datasets can provide a significantly improved performance when compared with the CNN trained on object-centric datasets for scene recognition tasks. Thus, we employed the Places-CNN, which is trained on 205 scene categories from the Places dataset with a total of 2.5 million images, to extract the features of roadworks and crosswalks images. Place-CNN has very similar architecture to Alex-Net, which is introduced in [4]; the detailed description relating to the training procedure of Places-CNN can be found in [3]. In addition to Places-CNN, we also want to investigate the performance of another pre-trained CNN called Hybrid-CNN; this CNN was trained on 205 scene categories of Places database, and 978 object categories of ImageNet with a total of 3.6 million images.

First, the original RGB images are resized to 227*227 pixels, so that they are compatible with the fixed size of the CNN input layer. Following this, the mean-subtracted images are propagated through the network from the input layer to the output layer; this step is also known as forward propagation. During forward propagation the activation responses of each layer are calculated. We extracted the FC7 layer activations from the CNN, owing to the fact that they represent the highest level features of images. The FC7 layer features are actually 4096-dimensional real-valued sparse vectors.

Having completed this step, the features of images were been successfully extracted, where classifiers such as SVM can be trained on them. We utilized the library LIBSVM [8] in order to train and test SVMs. LIBSVM take the data in the format "label index1:value1 index2:value2 ..." as input, for example "+1 1:0 2:0 3:0 ... 1591:0.0658973 ...". In certain instances, normalization or scaling of feature vectors might also be helpful. There are different types of SVMs, such as SVM with linear kernel or with RBF kernel. Initial experiments showed that RBF-SVM performed much more effectively than linear-SVM, therefore, the former has been utilized in subsequent experiments.

4.2. CNN Fine-tuning

Previous work, such as [6], demonstrates that the fine-tuning of a pre-trained CNN can yield significant accuracy improvements on the PASCAL VOC 2012 object detection task. However, we are not aware of any previous work that evaluates the efficiency of the fine-tuning approach on the scene recognition task. Our second approach for roadworks recognition is to fine-tune the Places-CNN on a more specific dataset for roadworks, with the hope that the fine-tuned Places-CNN can learn more effective features for roadworks images. We use Places-CNN as our pre-trained model, owing to the fact that we believe the parameters of Places-CNN can provide a competent starting point for fine-tuning.

Places-CNN is trained on the large, general Places database in a supervised fashion. We keep the architecture and parameters of Places-CNN aside from replacing its 205-way output layer (FC8) with a 2-way output layer; one way is for the yes-instances, while the other way for the no-instances. The parameters of the new FC8 layer are randomly initialized. After this, the CNN is trained (i.e. fine-tuned) using the stochastic gradient descent algorithm (SGD) on a small domain-specific dataset.

A necessary step before the actual training begins is to shuffle the training data: this is valuable insofar as it can help to reduce the risk that a sequence of corrupted training samples leads the searching procedure of SGD in a direction that is away from the local minimum. The training procedure begins with a base learning rate of 0.001, and the learning rate of the newly added FC8 layer is set to a value that is ten times larger than the base learning rate (i.e. the learning rate of other layers), which is 0.01, as it is randomly initialized while the other layers are already pre-trained. Moreover, we freeze the conv1 layer, since conv1 is relatively general. The mini-batch size is 100; the weights of the CNN are

updated in order to minimize the error on a mini-batch. The training procedure can take thousands of iterations, until the training loss of an iteration, which measures the cost of false predictions, reaches an acceptable level.

5. Evaluation

In this section we present the experiment setup to evaluate our approaches and also the experimental results. Subsection 5.1 compares the efficiency of the two approaches for recognizing roadworks. In order to explain the results on roadworks more effectively, in Subsection 5.2 feature visualization facilitates an intuitive understanding of the features learned by CNN. In regard to crosswalks, we only implemented the first approach (CNN+SVM), since Places-CNN has a default output node for the crosswalk category, no modification of the CNN architecture and fine-tuning is necessary.

5.1. Roadworks

5.1.1. Data. Both approaches are supervised learning, as a result, need labeled images as training data. We have selected about 12 thousand images from the original dataset containing approximately 70 thousand images extracted from the video. The number of roadworks and non-roadworks images are nearly equal; this can lead to more balanced recognition accuracy for roadworks and non-roadworks.

In the early experiments, 10%-20% of the selected 12 thousand images were used as test data. However, the recognition accuracy of the CNN+SVM approach can easily reach more than 99%, which is probably unpractical. A possible reason to this problem is that consecutive images are very similar; if one image appears in the training data, its neighboring images can be easily classified by the SVM. Therefore, for testing we selected 500 consecutive roadworks images and 500 consecutive non-roadworks images as the test data. This allows a reliable test owing to the fact that 500 consecutive images can cover a long distance. Moreover, the two approaches use the same training dataset and test dataset for a fair comparison between them.

5.1.2. CNN+SVM. After the FC7 layer features of all images in training dataset are extracted, it is often beneficial to normalize or scale the feature vectors. For normalization, we divide the feature vectors by their norms. Scaling refers to the process where each dimension of a feature vector is scaled to a given interval. Figure 2 shows the experimental results when different preprocessing steps are applied to the feature vectors before they are used to train a SVM classifier. Table 1 lists the most balanced recognition accuracies. As can be seen in the table, normalization or scaling of feature vectors can lead to the improvement of accuracy.

As a comparison with the features extracted by Place-CNN, we also carried out experiments to evaluate the efficiency of Hybrid-CNN features. As shown in figure 2, the recognition accuracy of the SVM based on Hybrid-CNN features is relatively unbalanced; for non-roadworks the accuracy can achieve nearly 100%, however, for roadworks, the accuracy is less than 40%, and the total accuracy is less than 70%.

	Roadworks	Non-roadworks	Total Accuracy
Places-CNN	0.82	0.842	0.831
Normalized	0.876	0.812	0.844
Scaled	0.854	0.828	0.841

TABLE 1: *Test results of the CNN+SVM approach using Places-CNN features, normalized Places-CNN features and scaled Places-CNN features.*

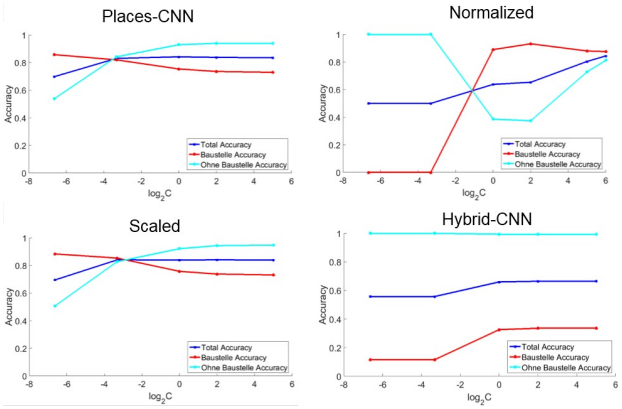

Figure 2: *Test results of the CNN+SVM approach.*

In figure 3 we present four incorrectly classified images when normalized Places-CNN features are used. The two images in the first row are false positives, namely roadworks images which are classified as non-roadworks, and the two images in the second row are false negatives, namely non-roadworks images which are classified as roadworks. As can be seen, there are images which are obviously roadworks however still have been incorrectly classified as non-roadworks. Thus, we think that the features learned by Place-CNN are probably not the most effective ones for some roadworks images. We have thought of two possible ways in which to account for this problem. This first possible reason is that training images are not clear enough. This could be attributed to that the resizing of images from the original 1280*640 to 227*227 pixels leads to information distortion, and that the images were taken under different illumination conditions. The other possible reason is that Place-CNN is trained on a general dataset, and not specifically trained for roadworks.

5.1.3. CNN Fine-tuning. For the CNN fine-tuning approach, we used just the same training data and test data as the CNN+SVM approach in order to achieve a fair comparison. Details of the training and test data can be found in section 5.1.2.

Figure 3: *False classifications of the CNN+SVM approach. In the first and second row are two false positives and two false negatives respectively.*

CNN training was performed on a GPU, and the entire training procedure ended after 3000 iterations in order to prevent overfitting. Since the mini-batch size was 100, and the training data had a total of ca. 12000 images, the CNN was trained for 25 epochs. For reference, we also trained a CNN with the same architecture from scratch; in other words, this CNN was randomly initialized. Figure 4 shows the training loss of both CNNs after each iteration during the training process. The pre-trained CNN began with a loss of 0.623, and the randomly initialized CNN with a loss of 1.123. After approximately two hundred iterations of weight updates, the training loss of the pre-trained CNN started to converge on less than 0.1, while the training loss of the randomly initialized CNN converged to some number near 0.7.

After the fine-tuning of Places-CNN was finished, the fine-tuned model was tested on the test dataset with 1000 images. Table 2 lists the test results. The randomly initialized CNN produced only one prediction for all test images, in spite of the fact that it was trained for 3000 iterations. Moreover, Places-CNN, before fine-tuning with the output softmax layer only being randomly initialized, also produced just one prediction for all test images. After 200 iterations of fine-tuning, the recognition accuracy of Places-CNN was able to achieve a very high-level (>95%). Therefore, it should be noted that the tests results not only suggest the efficiency of fine-tuning but also the importance of the pre-trained model.

Figure 5 shows two images that were incorrectly classified by Places-CNN after 3000 iterations of fine-tuning. Dissimilar to the CNN+SVM approach, most of the incorrectly classified images by the fine-tuned Places-CNN are relatively ambiguous.

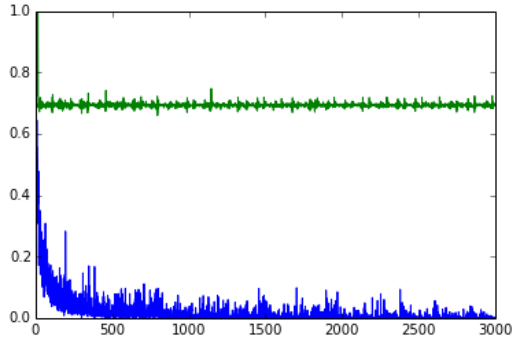

Figure 4: *Training loss of the pre-trained CNN and the randomly initialized CNN after each iteration of fine-tuning, the blue line indicates the training loss of the pre-trained CNN, the green one belongs to the randomly initialized CNN.*

	Roadworks	Non-roadworks	Total Accuracy
Scratch(3000iter)	0.0	1.0	0.5
0 Iteration	1.0	0.0	0.5
200 Iterations	0.944	0.998	0.971
400 Iterations	0.81	1.0	0.905
1000 Iterations	0.988	0.968	0.978
2000 Iterations	0.976	0.982	0.979
3000 Iterations	0.948	0.968	0.958

TABLE 2: *Test results of the fine-tuning approach. In the first row are the test results of the randomly initialized CNN after 3000 iterations of fine-tuning. The rest are results of the pre-trained Place-CNN after 0, 200, 400, 1000, 2000 and 3000 iteration(s) of fine-tuning. Fine-tuning of 0 iteration means no fine-tuning.*

5.2. Feature Visualization

In section 5.1, we demonstrated that the CNN fine-tuning approach can significantly outperform the CNN+SVM approach on the roadworks recognition task, and, as a result of this, we have reason to believe that fine-tuned Places-CNN has the ability to learn more effective features for roadworks. An easy way to prove our conjecture is feature visualization. Various previous work has been proposed to visualize the features learned by CNNs, for example [13] introduce the gradient-based visualization techniques, and [14] utilize the Deconvolutional Networks (deconvnet) to project the high-level features down to the

Figure 5: *Two false positives of Places-CNN after 3000 iterations of fine-tuning.*

pixel space of the input layer. In this practical course, we employed the deconvnet approach proposed by Zeiler et al. [14].

We utilized an implementation of the deconvnet approach on Github[1], the original version of Caffe was extended with a number of new types of layers; these included a "pooling switches layer", which allows max pooling to save the "switch variables", and so on. The first step was passing the input image to the well-trained CNN, where, following this, activations of all layers were calculated through forward propagation. In order to visualize the pool5 layer feature, we created a deconvnet and initialized it based on the parameters of the well-trained CNN. After this, the pool5 layer activations were presented to the deconvnet as input. With the help of "switch variables", input features were forward propagated through the deconvnet, and finally mapped back onto the pixel space.

Figure 6 shows the feature visualizations from the original Places-CNN and the fine-tuned Places-CNN after 3000 iterations. Through the reconstructions shown in column 1, 2 and 3, we are able to find that numerous road signs are overlooked by the original Places-CNN; however, they are still captured by fine-tuned Places-CNN. Reconstructions in column 4 and 5 reveal another advantage of the fine-tuned Places-CNN; that is, despite the fact that both CNNs can capture the signs of roadworks, the fine-tuned Places-CNN focuses more on the part related to roadworks in the image while the other part of image is almost ignored.

Figure 7 demonstrates the progression during the fine-tuning process. It should be noted that there is almost no difference between the visualizations from the fine-tuned Places-CNNs after 1000 and 3000 iterations. Moreover, still slight differences can be observed between the visualizations from the fine-tuned CNN after 200 and 3000 iterations. In light of this, we consider that the fine-tuning of 1000 iterations is already sufficient to provide outstanding performance for roadworks recognition, more iterations may not help CNN learn more effective features for roadworks.

5.3. Crosswalks

Dissimilar to roadworks, Places-CNN has an output node for the scene category crosswalk. Therefore, Places-CNN can be directly used to recognize crosswalks. Hence for crosswalks recognition, we only implemented the CNN+SVM approach and the fine-tuning of Places-CNN was not necessary.

After the experiments on roadworks, we collected some useful training strategies that can also be employed in experiments on crosswalks. As a result, we only utilized Places-CNN for feature extraction and the extracted feature vectors were normalized. We separately evaluated the performance

1. https://github.com/mylxiaoyi/caffe-deconvnet

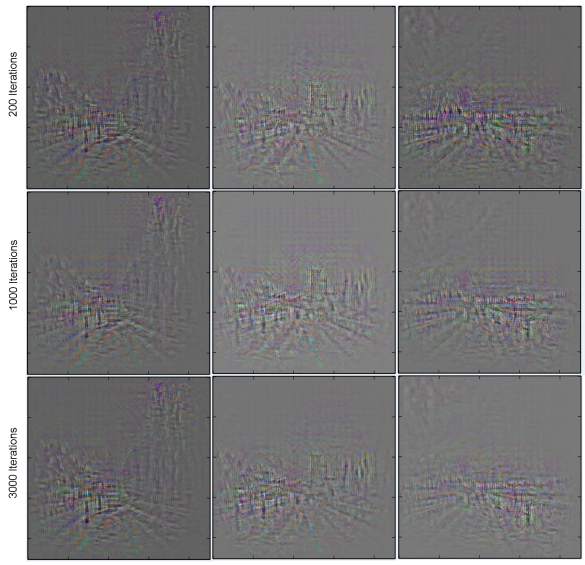

Figure 7: *The feature visualizations of three images. Each column presents the feature visualizations of a image. In the first, second and third row are the visualizations from the fine-tuned Places-CNN after 200, 1000 and 3000 iterations respectively.*

	Crosswalk	Non-crosswalk	Total
Training Data	119	127	246
Test Data	18	12	30

TABLE 3: *The statistics of the training and test data of SUN database.*

of the CNN+SVM approach on two datasets, the SUN and Places databases.

5.3.1. SUN Database. The SUN database is a large database for scene recognition, which contains 899 scene categories and 130,519 images in total [15]. We selected 105 images of crosswalks from SUN and randomly selected 108 non-crosswalk images from other categories. In addition to crosswalk images from SUN, we also selected 32 crosswalk images and 31 non-crosswalk images from the video. We randomly selected 30 images from all images to serve as test data; this comprised approximately 10% of all images. Table 3 lists the statistics of the training and test data.

Owing to the fact that Place-CNN can recognize a crosswalk directly, we can compare the performance of the CNN+SVM approach and Place-CNN. Experimental results are listed in table 4. It is important to note that the CNN+SVM approach has achieved exactly the same accuracy as Place-CNN, insofar as they both have only two false positives. Figure 8 shows the two incorrectly classified images of both approaches, and they have one common false prediction.

5.3.2. Places Database. The Places database is much larger than the SUN database. We selected 8511 crosswalk

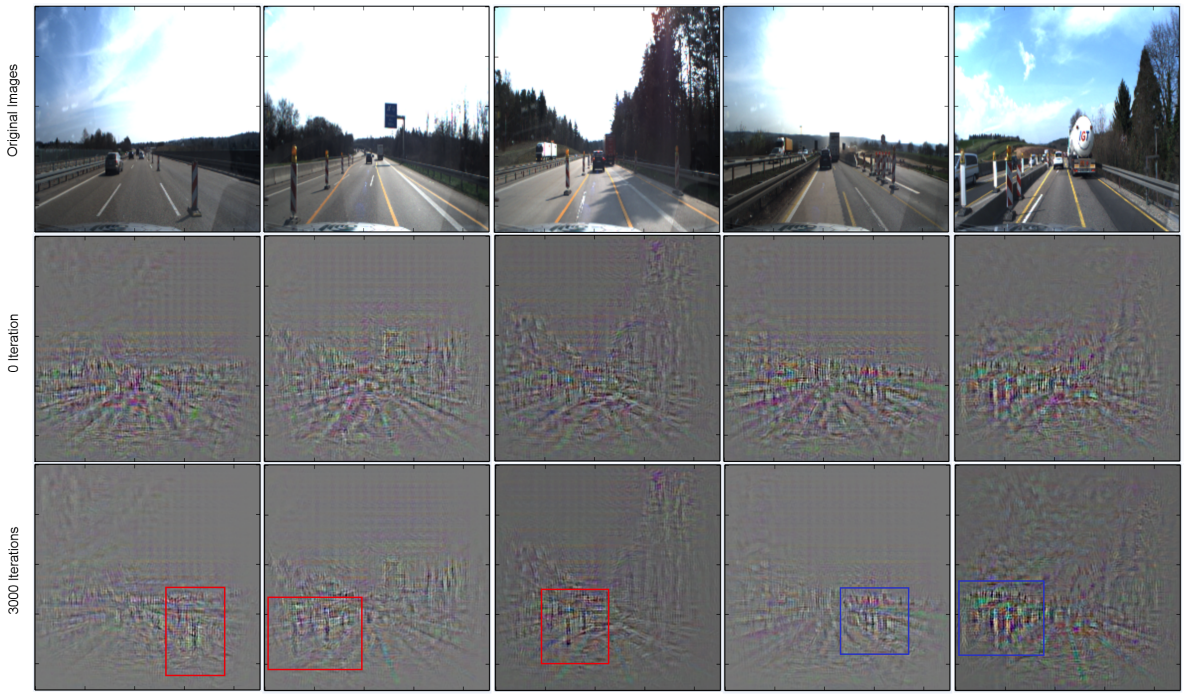

Figure 6: *The feature visualizations of five images from the original Places-CNN and the fine-tuned Places-CNN after 3000 iterations. In the first row are the original images, in the second and third row are the visualizations from the original Places-CNN and the fine-tuned Places-CNN after 3000 iterations respectively. The red boxes indicate the road signs which are overlooked by the original Places-CNN but still captured by the fine-tuned Places-CNN. The blue boxes indicate that the fine-tuned Places-CNN focuses more on the part related to roadworks in the image while the other part of image is almost ignored.*

	Crosswalk	Non-crosswalk	Total Accuracy
CNN+SVM	88.9% (16/18)	100% (12/12)	93.3% (28/30)
Places-CNN	88.9% (16/18)	100% (12/12)	93.3% (28/30)

TABLE 4: *Test results of the CNN+SVM approach and Places-CNN on SUN database.*

	Crosswalk	Non-crosswalk	Total
Training Data	7729	7282	15011
Test Data	782	718	1500

TABLE 5: *The statistics of the training and test data of Places database.*

images from Places and 8000 non-crosswalk images from the category highway. We still randomly selected approximately 10% of all images as test data. The statistics of the training and test data are presented in table 5.

Due to the fact that Places-CNN is also trained on the Places database, including those selected images, a comparison of the CNN+SVM approach and Place-CNN on Places database is certainly unfair. In light of this, we evaluated only the CNN+SVM approach on this database, and the evaluation results are presented in table 6. Figure 9 shows four false classifications of the CNN+SVM approach on Places database.

Figure 8: *Two incorrectly classified images of the CNN+SVM approach and Places-CNN on SUN database, and they have one common false prediction.*

6. Conclusion

In this work we have implemented two approaches for recognizing roadworks and crosswalks. For roadworks, the CNN+SVM approach reached an accuracy of nearly 85%, and the CNN fine-tuning approach reached an accuracy of over 95%. In order to explain the accuracy improvement facilitated by the fine-tuning, we implemented

	Crosswalk	Non-crosswalk	Total Accuracy
CNN+SVM	95.6% (748/782)	100% (12/12)	95.2% (1428/1500)

TABLE 6: *Test results of the CNN+SVM approach on Places database.*

Figure 9: *False classifications of the CNN+SVM approach on Places database.*

feature visualization using deconvnet. Experimental results demonstrated that the fine-tuned Place-CNN focuses to a greater extent on the signage of roadworks; and it is capable of learning the road signs of roadworks that are entirely ignored by the original Places-CNN. In this way, we demonstrated that, through fine-tuning, we can obtain a more effective model for roadworks recognition.

In regard to crosswalks, the best accuracy of the CNN+SVM approach is over 96%; we attribute this effective performance to the fact that Places-CNN was trained on crosswalks images, and, as a result of this, features learned by Place-CNN are highly discriminative. Furthermore, we demonstrated that the CNN+SVM approach performs well even for low resource, namely approximately 300 images.

In further work we hope to do more experiments on fine-tuned CNN features. We are curious as to whether or not a combination of fine-tuned CNN features and the SVM classifier could provide us with further performance improvement.

Acknowledgments

We would like to thank Prof. Zllner, our supervisors Michael Weber, Florian Kuhnt, and Marc Zofka for their valuable advice, continuous support during this practical course, and for providing us with GPU computing power at FZI.

References

[1] Fei-Fei, Li, and Pietro Perona. “A bayesian hierarchical model for learning natural scene categories.” Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 2. IEEE, 2005.

[2] Oliva, Aude, and Antonio Torralba. “Modeling the shape of the scene: A holistic representation of the spatial envelope.” International journal of computer vision 42.3 (2001): 145-175.

[3] Zhou, Bolei, et al. “Learning deep features for scene recognition using places database.” Advances in Neural Information Processing Systems. 2014.

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

[5] Donahue, Jeff, et al. “Decaf: A deep convolutional activation feature for generic visual recognition.” arXiv preprint arXiv:1310.1531 (2013).

[6] Girshick, Ross, et al. “Rich feature hierarchies for accurate object detection and semantic segmentation.” Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.

[7] Oquab, Maxime, et al. “Learning and transferring mid-level image representations using convolutional neural networks.” Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014.

[8] Chang, Chih-Chung, and Chih-Jen Lin. “LIBSVM: A library for support vector machines.” ACM Transactions on Intelligent Systems and Technology (TIST) 2.3 (2011): 27.

[9] LeCun, Yann, et al. ”Gradient-based learning applied to document recognition.” Proceedings of the IEEE 86.11 (1998): 2278-2324.

[10] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014).

[11] Szegedy, Christian, et al. ”Going deeper with convolutions.” arXiv preprint arXiv:1409.4842 (2014).

[12] Jia, Yangqing, et al. “Caffe: Convolutional architecture for fast feature embedding.” Proceedings of the ACM International Conference on Multimedia. ACM, 2014.

[13] Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps.” arXiv preprint arXiv:1312.6034 (2013).

[14] Zeiler, Matthew D., and Rob Fergus. “Visualizing and understanding convolutional networks.” Computer VisionECCV 2014. Springer International Publishing, 2014. 818-833.

[15] Xiao, Jianxiong, et al. “Sun database: Large-scale scene recognition from abbey to zoo.” Computer vision and pattern recognition (CVPR), 2010 IEEE conference on. IEEE, 2010.