Deep Integration: A Multi-label Architecture for Road Scene Recognition

Long Chen, *Senior Member, IEEE* Wujing Zhan, Wei Tian, Yuhang He and Qin Zou, *Member, IEEE*

***Abstract*—Deep convolutional neural networks have been applied by automobile industries, internet giants and academic institutes to boost autonomous driving technologies, while progress has been witnessed in environmental perception tasks such as object detection and driver state recognition, the scene-centric understanding and identification still remain a virgin land. This mainly encompasses two key issues: 1) the lack of shared large datasets with comprehensively annotated road scene information, and 2) the difficulty to find effective ways to train networks concerning the bias of category samples, image resolutions, scene dynamics, and capturing conditions, etc. In this work, we make two contributions. i) We introduce a large-scale dataset with over 110k images, dubbed DrivingScene, covering traffic scenarios under different weather conditions, road structures, environmental instances and driving places, which is the first large-scale dataset for multi-class traffic scenes classification; ii) we propose a multi-label neural network for road scene recognition, which incorporates both single- and multi-class classification modes into a multi-level cost function for training with imbalanced categories and utilizes a deep data integration strategy to improve the classification ability on hard samples. The experimental results on DrivingScene and PASCAL VOC demonstrate the effectiveness of the proposed approach in handling the challenge of data imbalance.**

***Index Terms*—Large-scale dataset, data imbalance, multi-label classification, road scene recognition.**

I. Introduction

Self-driving has received vast capital inflow and tremendous research interest in both academia and industry in recent years. Researchers coming from various global esteemed institutes and top-tier mobile manufactures, join together to push the boundary of self-driving challenges. Self-driving technology can be divided into several parts, including localization and mapping [1]–[3], motion planning [4], [5], behavioral decision [6], [7], and scene understanding [8], etc. While the first three parts have already been extensively researched, scene understanding has not been well studied or solved. The primary two reasons for this include the lack of publicly available large-scale scene-centric dataset in self-driving domain and the lack of effective training approach on these datasets to deal with data imbalance brought by category samples and image resolutions.

The corresponding author is Long Chen.

L. Chen, W. Zhan and Y. He are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, P.R. China. (Email: chenl46@mail.sysu.edu.cn; zhanwj5@mail2.sysu.edu.cn)

W. Tian is with the Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany. (Email: wei.tian@kit.edu)

Q. Zou is with the School of Computer Science, Wuhan University, Wuhan, Hubei, P.R. China. (Email: qzou@whu.edu.cn)

Generally, scene understanding involves a number of subtasks such as scene categorization, object detection/tracking, and semantic segmentation, etc. Each of these tasks describes a particular aspect of a scene. It is very hard to jointly model some of these aspects to exploit the relations between different elements of the scene and obtain a holistic understanding. Innovative investigation and novel solutions are in great demand. Deep convolutional neural networks (DCNNs) have achieved great success in a number of computer-vision tasks such as image classification [9], [10], object detection [11], [12] and tracking [13]. Some recently-presented architectures even allow for per-pixel predictions like semantic segmentation [14], [15] or scene-flow estimation [16]–[18]. Hence, it would be reasonable to try and develop some DCNN-based technologies for scene classification.

On the other hand, DCNNs are highly dependant on the training dataset to obtain a high performance. Challenging datasets often play an important role in validating the performance of advanced deep models as well as in stimulating new algorithms. For example, the ImageNet LSVRC-2010 dataset, which contains 1.2 million high-resolution images of 1,000 different classes, has given a great help in discovering the capacity of DCNN models in image classification [19]. Another successful case is the dataset of Places-Standard [10] which has about 1.8 million images from 365 scene categories with at most 5,000 images per category. Deep networks trained on this dataset exhibit an excellent performance at scene recognition. Although the combination of various CNNs [20], [21] and different large image datasets [9], [10], [22] has gained great achievements on object classification and large area scene recognition. However, as most of them are focusing on recognition of natural scenes, the object-centric features learned by deep neural networks often lack of richness and diversity and thus are insufficient for understanding complex traffic scenes.

In this paper, we first introduce a largescale dataset, called DrivingScene, which is initially designed for large-scale scene recognition in self-driving domain. It comprises over 110K images with common traffic scenes collected by both vehicle dash camera and web search engine, including different weather conditions, road structures, environmental instances and driving places, with fine-grained and carefully annotated labels. We make sure that each class has a training set of more than 400 samples, which is close to the challenge standard of LSVRC. We hope our DrivingScene dataset can help to learn richer traffic scenes.

Although object classification and scene classification have made great achievements in ImageNet and Places datasets,

there are still some problems unsolved. For example, there are some erroneous labels among the 280 million labeled images of ImageNet, and there exist a lot of ambiguous and similar semantic categories in the label corpus of Places. All these points have brought greater challenges to the training of deep networks. While in our DrivingScene, fine-grained and carefully annotated labels have been provided to avoid the above problems. Comparatively, the main challenges of the DrivingScene dataset lie in the following points:

• Multi-class prediction. This dataset provides multiple labels for the road scenes. Hence, the road scene classification will be featured by recognizing multiple categories of objects at the same time.

• Data imbalance. This dataset holds a vast variety of traffic scenarios. However, the scale of each category is not the same, and some rarely appeared scenes have fewer images. Therefore, how to train the deep convolutional networks to ensure a high accuracy on small categories is a very challenging problem.

• Varied image resolution. All images in this dataset are captured in the real world. However, the data source includes both the Internet and the real vehicle driving. It is difficult to maintain a uniform resolution for them. Hence, it requires the network to be able to suppress the influence of varied image resolution of the inputs.

Regarding the above points, in this paper, we also introduce a new multi-label neural network as a baseline on the DrivingScene dataset. The architecture exploits hybrid-labels which includes both multi- and single-labels. The multi-labels are mainly used for multi-category prediction learning while the single labels are used to enforce the supervised-learning of hard samples or small categories which need to be more carefully handled during the training procedure. Additionally, we propose a deep data integration method, which uses a boosting method to guarantee an adaptive sampling of scene images, especially for imbalanced category samples. As image quality varies with respect to the compression approach, we further employ resolution-adaptive procedure in our network to improve the robustness against the noise from varied image resolutions.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III introduces the DrivingScene dataset. Section IV describes the proposed multi-label scene recognition network. Section V reports the experiments and results. Section VI concludes our work.

II. Related Work

In this section, we present researches related to multi-label scene recognition from two aspects: the dataset of self-driving scenes and the classification with biased data.

Dataset of self-driving scenes Large amounts of labeled dataset are often iconically likened as the fuel to deep learning rocket, without which can the tremendous progress of vision based research less likely be achieved. Taking object recognition for example, ILSVRC uses a subset of ImageNet [9] with roughly 1,000 images in each of 1,000 categories, and the COCO dataset [22] provides nearly 120,000 training samples with a total of 80 categories. Both of them provide a large number of training samples to guide the convolutional network to achieve high recognition rate. Other examples for large training set-benefitted scene recognition can also be found, e.g., the SUN dataset [23] provides a wide coverage of scene categories containing 397 categories with more than 100 images per category, the Places [10] contains 1.8 million images from 365 scene categories, with at most 5,000 images per category, etc.

However, there is no such dataset in the autonomous-driving field. Some traffic-scene datasets mainly focus on environmental perception, with the self-driving scene recognition almost neglected. For example, KITTI [24] comprises a wide range of challenges like stereo vision, odometry, object detection and tracking, *etc*. CompCar dataset [25] specifically focuses on fine-grained car classification/verification and attribute prediction. CityScapes [8] provides a large-scale dataset derived from stereo sequences, aiming at both pixel- and instance-level semantic labeling. The self-driving scene recognition is simple for human brains but extremely difficult for computers to address. To attract and motivate more research on self-driving scene recognition [1], we thus introduce a new large-scale dataset of over 110k scene images cutting across 52 categories, which is currently, to our best knowledge, the largest dataset in terms of scene recognition in self-driving domain. The most related work to ours is the FM2 dataset [26], yet it contains a total number of 6,237 images from eight scene classes.

Classification with biased data In scene recognition, a single image usually associates with multiple scene labels. Thus, most prior works train a deep neural network to assign the multi-class label to the query image [27]–[30]. Although some deep structures such as VggNet [31], Inception V3 [32] and ResNet [33] have demonstrated higher performance with deeper layers in classification, the training still suffers from the negative impact of data imbalance. Attempting to solve this problem, two approaches are well studied in past years. The first approach is re-sampling, adopted by Oquab *et al*. [34] to rebalance class priors during training through under- and over-sampling. The second is the cost-sensitive learning, utilized by Huang *et al*. [35] along with the Triple-Header Hinge Loss to assign different costs for misclassification on the majority and minority classes. Despite a good performance, both methods are proposed mainly for single-label classification. Moreover, the over- and under-sampling may introduce undesired noise or remove valuable sample information while the cost-sensitive learning usually requires utilization of additional features.

The above works can be seen as natural extensions to the existing imbalanced learning techniques, while neglecting the underlying data structure for discriminating imbalanced data. To explore a more effective way to deal with data imbalance in the context of deep representation, we incorporate both single- and multi-label training by a multi-level loss function, making our architecture more flexible and compact. By deploying a deep data integration method, we re-balance class priors

[1]We are planning to publish this dataset to push forward the research on traffic scene recognition.

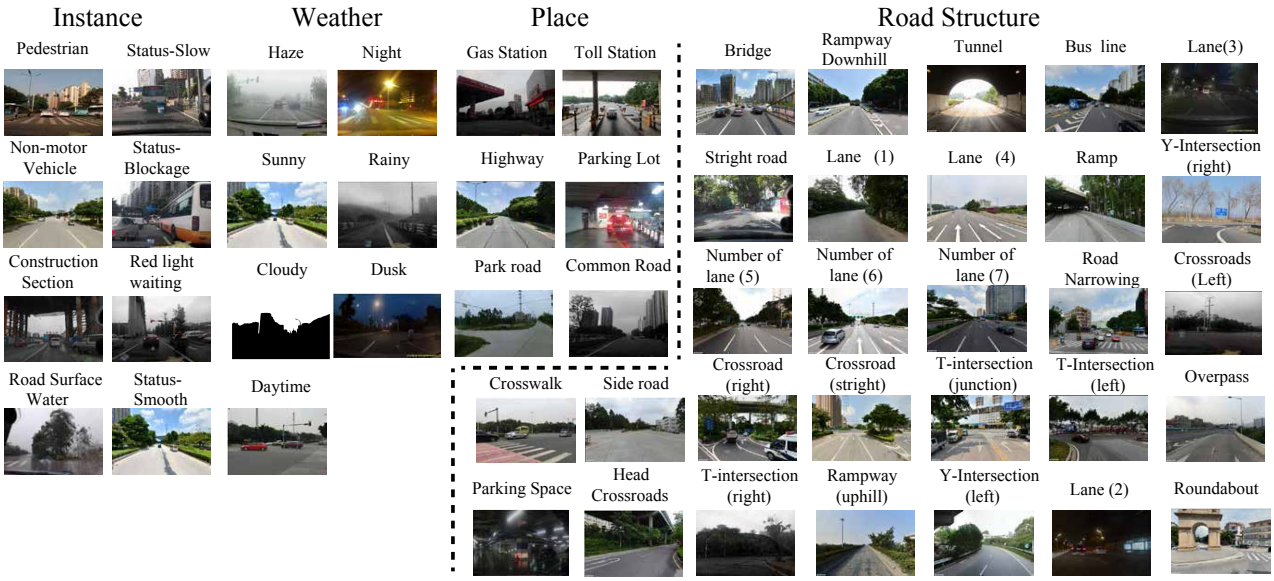


Fig. 1: The DrivingScene dataset. Subclasses of each super class are displayed together.

while emphasizing the cost for misclassifications based on a combination of two classic schemes – data resampling and cost-sensitive learning. Two methods [36], [37] in solving the problem of imbalance are close to ours. Both of them combine multiple methods to boost the effectiveness of the model. For example, in [36], Kumar *et al*. introduced a novel cascaded architecture that benefits from both ensemble and hard sample mining techniques. They train a convolutional network at multi-level and on each level conduct re-sampling according to the error cost of the last level. In comparison, the proposed approach models complex dependencies between class labels and benefits from the training strategy of boosting methods. In [37], Huang *et al*. designed a new sampling method named Quintuplet sampling. It combines with a triple-header loss to learn discriminative features. In contrast, our approach directly improves the learning ability of the network in an end-to-end manner.

Another problem is how to handle images in multiple resolutions. A typical approach is the multi-scale cropping, adopted by the VggNet [31] and then inherited by following works, such as ResNet [33] and Inception V3 [32]. Aside from that, in other works, the multi-scale representation is preferred. For instance, Oquab *et al*. [34] propose a new re-sampling method about multi-scale part proposals for fine-grained categorization. Wang *et al*. [38] combine multi-resolution architecture with a confusion matrix for scene classification. However, the cropped patch may lose the label information associated with other image regions and additional training work is required to learn networks for multiple resolutions or scales.

III. Deep DrivingScene Dataset

To maximumly strengthen the robustness and completeness of our deep DrivingScene dataset, we fully take geographical location, temporal variation, static and dynamic characteristics into consideration when collecting the dataset. Besides, we define each driving scene by combing the world-wide transportation construction rule and human-biased understanding towards all scenes. In sum, we provide 52 different kinds of driving scenes, cutting across common driving instances, weather conditions, places and road structures. The scene category structure is shown in Fig. 1, in which each image is tagged with fine-grained and carefully annotated labels.

A. Scene Corpus

driving instance Driving instance here indicates the instances currently happening on traffic roads and they are temporally short, including traffic congestion, road construction, red light waiting, pedestrian crossing and road with surface water. In terms of traffic congestion, we further classify it into three status according to the congestion level: smooth, slow and fully-blocked. We involve pedestrian crossing and road with surface water in traffic instance because both of them represent short traffic state.

driving place Place are static driving scenes that are spatially distributed under various road conditions. It can be divided into three broad categories. First, traffic place indicates road categories, including highway, common road, park road. Second, it contains various public services, for example, toll station, gas station, parking lot, public transportation station, *etc*. Third, traffic place features road static characteristics, including crosswalk and parking space.

weather condition We firmly believe that a robust driving scene dataset or an elegant scene recognition algorithm has to deal with different weather conditions. Any self-driving technology has to pass harsh weather test before its deployment for real applications. In terms of driving scene recognition, involving the same driving scene but with different weather intervention helps algorithms to learn deep discriminative features. To this end, we consider 4 common weather conditions, namely sunny, cloudy, rainy and haze. Furthermore, we add

(1)Instance: Red light waiting (2)Place: Park Road (3)Weather

(4)Road Structure: T-Intersection (5)Road Structure: Lane Number(2/3/5/6) (6)Road Structure: Y-Intersection

Fig. 2: Sample images of DrivingScene dataset.

Dataset	Type	Traffic Classes	Scene samples	Training images per class	Different time tag	Different weather tag
SUN [23]	Single-label	19	1629	47	✗	✗
Places [10]	Single-label	17	550k	1553	✗	✗
KITTI [24]	Multi-label	18	200	8	✓	✗
Cityscapes [8]	Multi-label	18	25k	442	✓	✗
DrivingScene	Multi-label	52	110k	400	✓	✓

TABLE I: Comparison of different datasets. The check mark indicates that the mentioned tag is provided.

light change factor to our dataset and choose three particular time spots: daytime, dusk and night.

road structure road structure is the most essential part in self-driving as it directly guides self-driving vehicles' control and path planning. Besides, road structure serves as the direct medium for self-driving environment perception. We divide the road structures into four subcategories: road lane, road intersection, road trend and specific roads. Road lane has been extensively researched in self-driving, road lane detection, tracking, keeping and changing provides important clue for self-driving system to make appropriate decision. We thus discriminate the road lane driving scene w.r.t. the lane number, which is in a range from 1 to 7 in our driving scene dataset. As to the intersection road structure, the "L-", "T-", "Y-" and "X-" like intersections are included with each type further containing sub-labels of turning left, turning right and going straight. In the road trend, road structure describes the upcoming road situation, including ramp way (downhill and uphill), road narrowing and U-turn. In the end, specific roads here indicate discrete and particular roads, for instance, overpass, tunnel, bridge and side way.

Samples are shown in Fig. 2, which provides a clear and intuitive understanding about the scene corpus in DrivingScene dataset. By following this structure guide, we start collecting the dataset.

B. Data Collection Procedure

We collect our deep DrivingScene dataset under two natures: vehicle on-board dash camera nature and website search nature. In the first one, a dash camera is mounted just behind the windshield and near the rear view mirror. While the vehicle drives around, it records various street scenarios with different daytime and weather conditions. The video images with camera shakes or large occlusions are directly filtered out. Furthermore, to reduce the scene overlaps, we only select one frame from every consecutive 30 frames. With the filtered dataset, professional labelers annotate each image with appropriate tags in the scene corpus by the tool we designed for multi-label tagging. Firstly, labelers are asked to label each image with as many tags as they can, thus to keep the labeling completeness. After that, labelers are further asked to double-check the labeling results from each other to guarantee the labeling accuracy.

On the world-wide web, we search the scene images with the key scene words using three different searching engines, namely the Google, Yahoo and Bing. The key scene words are

displayed in Fig. 1. The scenes with consistent scene words are classified into the same category. All the retrieved images are crawled and the bad ones or those with too small sizes are directly filtered out. Labelers take a further check of these retrieved images by archiving them into their relevant scene categories and deleting irrelevant ones.

The statics of DrivingScene dataset is given in Table I. It can be clearly observed that, comparing with other datasets, ours has three advantages: 1) we extend the recognition task into multi-category which is similar to benchmarks such as Cityscapes [8] and KITTI [24]; 2) we provide more scenario categories while ensuring that at least 400 images are included in each category for training; 3) we guarantee that images for each scene are collected in different time and weather, which largely enhances the challenge of classification since the existing deep models are sensitive to the variation of lighting, texture, and view angles. The second and third points indicate that, the DrivingScene dataset is rich in terms of density and diversity. The third point would also bring benefits for improving the generalization power of deep convolutional networks. We will discuss these points in detail in the following sections.

C. Diversity and Density Discussion

Publicly available image datasets are task dependent. And it is difficult to fairly compare them in terms of whether a particular dataset is better or worse, although the image number and category coverage range are often regarded as two import metrics. Here we argue that dataset diversity and density are two indicators for large-scale image dataset evaluation (*i.e.*, the Places dataset [10]), especially for deep feature learning tasks. Density is equivalent to data concentration, measuring the similarity level of an image with its neighbors. A dataset with higher density often guarantees to learn powerful representative features through deep convolutional neural networks (CNNs). Image statistics of our DrivingScene dataset is given in Table I, from which we can clearly observe that there is no large image number discrepancy among different scene categories. Furthermore, we strike a balance in image number between dash camera nature and website search nature, in which about 60% are captured in real scenes and the rest is collected from the internet.

The high density alone can deteriorate the dataset quality. An extreme situation is that all the images regarding a scene category are taken within the same viewpoint or with less camera pose variability. This high overlap of image appearance leads to large dataset redundancy, inevitably jeopardizing the algorithm's performance. Thus, a good dataset should have strong generalization capability and involve as many diverse images as possible regarding a scene category. We maximize DrivingScene dataset diversity from three aspects. First, we insist to capture images of the same scene category at different locations. For instance, in Fig. 2 (2), the image of scene "park road" are captured at various locations and they share large visual difference. Second, images of the same scene category are captured under different temporal conditions. For instance, in Fig. 2 (3), the scene images of single lane road structure are captured under night, haze, rainy and sunny weather condition. In the end, we involve large pose and viewpoint variability in DrivingScene dataset. For example, in Fig. 2 (4), the "T-" intersection road structure scene is captured with large back-and-forth pose variability and viewpoint gaps. The same collection process in the "Y-" intersection can be seen in Fig. 2 (6).

IV. Multi-label Scene Recognition

Generally, the scene recognition problem can be casted as training a model G_M, given a query image I_i from dataset $T = \{I_i | 1 \leq i \leq N\}$, to retrieve its label $\mathbf{y}_i$. In multi-label classification tasks, the label $\mathbf{y}_i = [y_{i,1}, \ldots, y_{i,K}]$ is usually a sparse binary vector with its element $y_{i,k}$ set to 1 if the corresponding image I_i is tagged with class k. The dimension K indicates the total class number of dataset (here we have $K = 52$). The estimated label of the model is hereby denoted as $G_M(I_i) = \mathbf{p}_i$.

Previous works have shown that the multi-label classification can be transferred into single-label classification problems [34]. Therefore, the imbalance of categories can be solved by over- and under- sampling of corresponding training samples. In the study of [38], the network can be trained in two ways: iteratively alternating between different class labels or firstly encoding small labels into super classes and thereafter trained hierarchically. However, as the dependency between labels is a problem, the over- and under- sampling approach can not use this relation while method [38] needs more knowledge from those dependencies to improve super class clustering.

Regarding these issues, we propose a hybrid approach which incorporates the single-label training procedure into the multi-label architecture. With its help, we are able to solve the imbalance between multiple categories while guarantee a high classification accuracy. The detailed structure can be seen in Fig. 3. The left side (separated by the dash line) is the proposed AdaBoost data layer, and the right side is the main network architecture. The data layer weights more on minority classes and misclassified samples for extracting more strong features. Details about this approach are described in following subsections.

A. Multi-label Architecture

The proposed architecture is based on GoogleNet [39]. We modify the network structure by adjusting the loss function $L_\sigma(\mathbf{y}_i, \mathbf{p}_i)$ in three loss layers to tackle imbalanced classification problems. In the first two layers, we choose the loss function $L_{softmax}(y_{i,k}, p_{i,k})$ to train a single label tag k, which is selected by a weighted random process, where big values are assigned to small classes. The weight is determined by the distribution of samples within each class, which is sorted into three groups: small tag group cls_{small} of sample number less than 1000, medium group cls_{med} with sample number between 1000 and 10000, and big group cls_{big} with more than 10000 samples. The class number within each group is respectively 17, 22 and 13.

In the third loss layer, we choose $L_{sigmoid}(\mathbf{y}_i, \mathbf{p}_i)$ as the multi-label loss function, which is the same as the output layer of original GoogleNet. All these modifications are illustrated

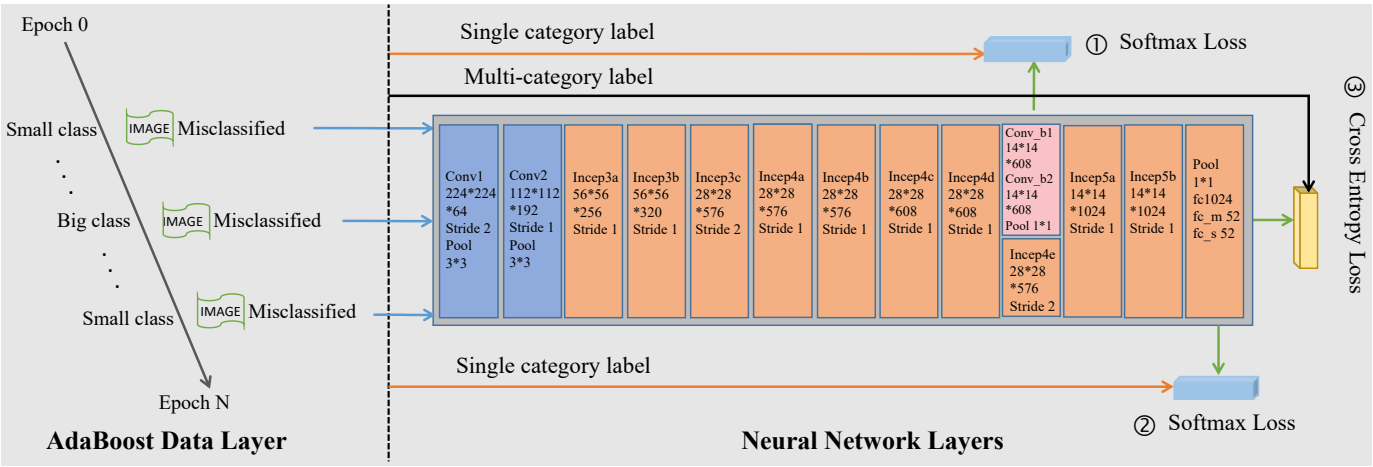


Fig. 3: Proposed network architecture. On the left side (separated by the dash line) is the AdaBoost data layer and on the right is the main network architecture with three loss layers, indicated by circled numbers. Due to exploitation of the AdaBoost data layer, hard samples are given more attention. The hard samples consist of small-class samples and misclassified samples. The convolution operation, pooling and the fully connection layer are abbreviated as Conv, Pool and fc, respectively.

in Fig. 3. In the GoogleNet architecture, an auxiliary function is used to help propagate the gradient to the lower layers, while in our architecture, one step further is taken which incorporates both single- and multi-class classification modes into a multi-level cost function for training with imbalanced categories. In our new architecture, each loss layer is placed after a fully connected layer and at least 4 pooling layers with the consideration that, by sufficient convolution and pooling operations, the learned feature should be more discriminative.

We utilize such kind of architecture which simultaneously feeds both single- and multi-category label into the network based on two reasons. On the one hand, because the loss function $L_{sofmax}(y_{i,k}, p_{i,k})$ is utilized in the single label training, the gradient descent direction can be forced to focus on the hard samples, according to the adjusted weight distribution obtained by an additional data integration method, as described in Section IV-B. On the other hand, as the single label tag is provided by the weighted random process and small classes are more probable to be chosen, the training on small label groups is also enforced. Usually, collecting images from these small scene categories is expensive, *e.g.*, in road scenes with extreme weather conditions. Therefore, the network requires to learn more features to classify these abnormal scene images, which is enabled in our approach. Finally, as our network is built on a multi-label architecture, simultaneous prediction of multiple scene labels is also granted.

B. Deep Data Integration Method

In above discussions, there is still one question left unsolved, which is how to effectively conduct sampling for selected single-labels. Intuitively, following the weighting strategy introduced in Section IV-A, more samples from small classes can be chosen during the training procedure. However, as the weight values are fixed, it is more likely to make the network overfitting on small tag groups. Hence, by incorporating a data integration method, we propose a self-adaptive sampling approach to train the network.

One of the most successful data integration methods is the AdaBoost algorithm [40], which iteratively adjusts sample weights to force the classifier to focus on classification errors. Analogously, we adapt the AdaBoost algorithm in an additional data layer to manage the sampling process, keeping the classification balanced between multiple label tags. As misclassified samples are with higher probability to be chosen, the network is more generalized in recognition of various traffic scenes.

In the data management layer, sample weights are firstly initialized with an equal distribution as $w^m_{sp,i} = \frac{1}{N}, 1 \le i \le N$ with superscript m to indicate the epoch number. After the network is finished training in current epoch, we calculate the sample error rate e^m_{sp} by

$$e^m_{sp} = \sum_{i=1}^{N} w^m_{sp,i}(G_M(I_i) \neq \mathbf{y}_i), \tag{1}$$

which equals the accumulated weights of misclassified samples. This term will be utilized to update sample weight in next epoch $m+1$ by

$$w^{m+1}_{sp,i} = \frac{w^m_{sp,i}}{Z^m_{sp}} \exp(\alpha^m_{sp}(G_M(I_i) \neq \mathbf{y}_i)) \tag{2}$$

where

$$\alpha^m_{sp} = \log \frac{1 - e^m_{sp}}{e^m_{sp}} \tag{3}$$

is the penalization factor and

$$Z^m_{sp} = \sum_{i=1}^{N} w^m_{sp,i} \exp(\alpha^m_{sp}(G_M(I_i) \neq \mathbf{y}_i)) \tag{4}$$

is the normalization parameter. Thus, the weight of misclassified samples will be scaled by a factor of $\frac{1-e^m_{sp}}{e^m_{sp}}$ under the assumption of $e^m_{sp} < 0.5$, which is valid in our experiments.

Since our network architecture consists of hybrid loss layers, for a better performance, we utilize an additional weighting

procedure, *i.e.*, the class weight $w^m_{cls,k}$. Unlike sample weights $w^m_{sp,i}$, the weight $w^m_{cls,k}$ is initialized according to the distribution of samples in each class k. For simplicity, we only initialize three different weight values according to the class groups defined in Section IV-A, thus we have

$$\begin{cases} W_1 = & \sum w^0_{cls,k}\ , \text{ for } k \in cls_{small} \\ W_2 = & \sum w^0_{cls,k}\ , \text{ for } k \in cls_{med} \\ W_3 = & \sum w^0_{cls,k}\ , \text{ for } k \in cls_{big} \end{cases} \tag{5}$$

subject to

$$\sum_{j=1}^{3} W_j = 1, 0 < W_j < 1. \tag{6}$$

Here we empirically set the positive constant W_1, W_2 and W_3 respectively to $0.89, 0.1, 0.01$. Under each class group we assume an equal distribution. After one epoch training, we calculate the class error rate by

$$e^m_{cls} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{i=1}^{N} w^m_{cls,k}(p^m_{i,k} \neq y_{i,k}\ \&\ y_{i,k}=1), \tag{7}$$

where $p^m_{i,k}$ is k-th element of estimated label vector $\mathbf{p}^m_i$ and N_k is the sample number of class k. Thereby, the update factor equals $\alpha^m_{cls} = \log\frac{1-e^m_{cls}}{e^m_{cls}}$ and the class weight is updated by

$$w^{m+1}_{cls,k} = \frac{w^m_{cls,k}}{Z^m_{cls}}\exp(\alpha^m_{cls}(w^m_{cls,k} < \frac{1}{K}\sum_{k}^{K} w^m_{cls,k})) \tag{8}$$

with normalization factor Z^m_{cls}. The class weights are embedded into the random process to determine which scene classes can be fed into network in the next epoch. The weighted random process is defined as

$$k^m_{max} = \arg\max_k(\text{rand}(1) + w^m_{cls,k}), \tag{9}$$

where process $\text{rand}(1)$ generates a random value with equal distribution over the range from 0 to 1. Index k^m_{max} is the single class label which should be fed into the network and such picking process is repeated for 40000 rounds in each epoch. The updated sample weight $w^m_{sp,i}$ is utilized for resampling the sample k in the training set T by a number of ceil$(w^m_{sp,i})$, Accordingly, the probability of misclassified samples chosen by the network is increased. Details about the training procedure is depicted in Alg. 1.

C. Resolution-adaptive Mechanism

Aside from the imbalance between multiple classes, another problem emerging at the training procedure is the varied size of input images. Fig. 4 presents the statics on image size of our dataset, ranging from a resolution of less than 1M pixels to over 12M pixels. Exactly, over 60% of the samples are with the same size and less than 1M pixels, while the remaining samples have randomly varying size and are unbalanced in the number. However, a fixed image size is required by the network, especially by the fully connected layers. Although this can be achieved by cropping the image into unified patches, as a scene label may only be characterized by specific image areas, the cropped image patch can loose the label property, resulting in unrecognized false positives. Another conventional approach is to resize the images. However, it would be difficult for the network to learn scene classes within shrinked image areas.

As to the feature levels, commonly, two approaches can be deployed to generate feature maps of fixed size for fully connected layers: the ROI [41] and SPP [42] pooling procedure. The location assumption from a given category, such as the label belonging to the weather category is assigned to the top area of the image. The first method cuts feature map of the

Algorithm 1 AdaBoost at training network.

1: **Input**:
- Convolutional neural network G^0_M.
- Training data set T.
- Maximum epochs $Epoch_{max}$.
- Epoch counter m.

2: **Output**: trained model G_{M^*}
3: **Steps**:
4: Initialization
   a) Set $m = 1$ and load the network G^{m-1}_M.
   b) Initialization of sample weights by $w^0_{sp,i} = \frac{1}{N}, 1 \leq i \leq N$.
   c) Initialization of class weights $w^0_{cls,k}$ w.r.t. Eq. 5.
5: While $m \leq Epoch_{max}$
   1) Load w_{cls} in the data layer of G^{m-1}_M. Train G^{m-1}_M within epoch m and save it as G^m_M.
   2) Test G^m_M on dataset T and calculate the sample error rate e^m_{sp} according to Eq. 1. The class error rate e^m_{cls} is calculated according to Eq. 7
   3) Update w^m_{sp} and w^m_{cls} w.r.t. Eq. 2 and Eq. 8.
   4) For each unreplicated sample I_i, resample it by a factor of ceil$(N(I_i) \cdot \frac{1-e^m_{sp,i}}{e^m_{sp,i}})$, where $N(I_i)$ is the replication number of sample I_i in dataset T.
   5) Update $m = m + 1$.
6: End
7: $G_{M^*} = G^m_M$.

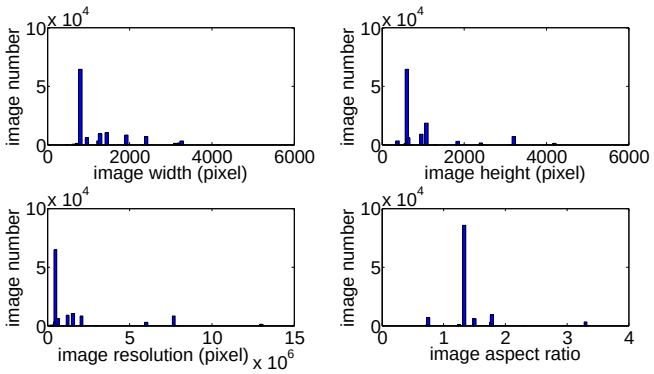


Fig. 4: Statistics on image size of our dataset. Sub-figures from top left to bottom right respectively show the distribution of width, height, total resolution and aspect ratio of images in our dataset.

consistent location into finely divided grids, and it performs the pooling operation in each grid, which is yet equivalent to cropping, loosing label property of the whole image [43]. The second approach builds a feature pyramid and conducts pooling in different scales. However, image information can not be equally represented in each scale and the scale number should be chosen carefully.

Different from those methods, the pooling procedure in our approach can be adapted to varied image resolutions. For instance, in the naive GoogleNet, also the basic architecture of our network, requires an input image size of 448×448 pixels. After processing by the previous net layers, the feature map is converted into a size of 14×14 pixels. For clarity, here we omit the channel number. In the next $pool5$ layer, a square pooling window of 5×5 pixels is shifted over the feature map by striding of 3×3 pixels. This yields a new feature map of 4×4 pixels, which is as the input of the fully connected layer. If we vary the input image resolution, *e.g.*, by 672×1152 pixels, the size of feature map before and after the $pool5$ layer is thereby changed into 21×36 and 6×11 pixels. However, we insist to generate the size-fixed feature map for the fully connected layer. Therefore, we modify the pooling window size and the striding step respectively into 7×12 and 4×7 pixels, which is illustrated in Fig. 5. This modification can be automatically performed by the resolution-adaptive mechanism:

$$\begin{aligned} sliding\ window\ size &= (\frac{5 * w}{14}, \frac{5 * h}{14}) \\ striding\ step &= (\frac{3 * w}{14}, \frac{3 * h}{14}) \end{aligned}, \tag{10}$$

where (w, h) indicates the feature map size before the $pool5$ layer and all the values in the window and step size are rounded to the next smaller integer. This pooling procedure can be applied onto images with varied resolutions. Hence, we obtain the feature map from a reshaped receptive field, without violating the fixed size requirement while maintaining the label property for the whole image.

Our method differs from SPP [42] in two points. One point is that, the sliding window for extracting features utilized by our method is more flexible than SPP, in which the window is usually with a square size. The other point is that our approach targets at processing biased image size and improve the classifier performance for small images without changing their original resolutions. However, the SPP targets at learning features in different scales under various resolution of the object.

V. Experiment and Evaluation

In this section, we first evaluate the diversity and density of the shared categories of SUN, Places and DrivingScene. Second, we give an ablation study on combining the proposed multiple softmax cross-entropy, deep integration and the resolution-adaptive mechanism with three backbones, *i.e.* VggNet, GoogleNet and ResNet-50. The proposed method is compared with four other methods that can also deal with imbalanced data. Third, we test the proposed method on Pascal VOC to validate its effectiveness on other datasets. Finally, we examine the differences of ImageNet, Places and DrivingScene by visualizing the neural responses of various network levels.

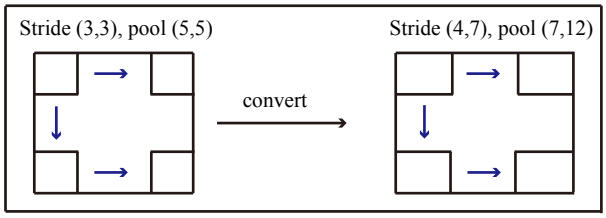


Fig. 5: The left side is a 14×14 feature map yielded by an image of 448×448 pixels. After being fed into the $pool5$ layer of GoogleNet, the output is a shrinked feature map of 4×4 pixels (we omit the channel number for clarity). On the right is another feature map in a size of 21×36 yielded by an image of 672×1152 pixels. To achieve the the same size of final feature map, *i.e.*, 4×4, we respectively modify the pooling window and the stride, by the resolution-adaptive mechanism, into 7×12 and 4×7 pixels.

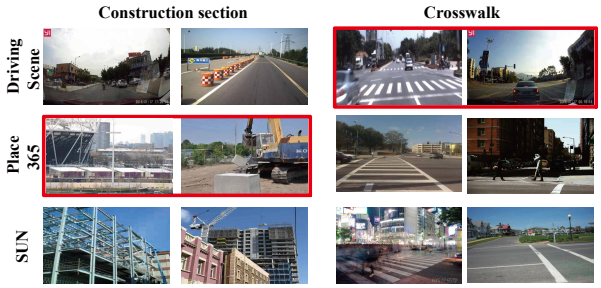


Fig. 6: Sample images of diversity evaluation. The most similar pair is highlighted in red.

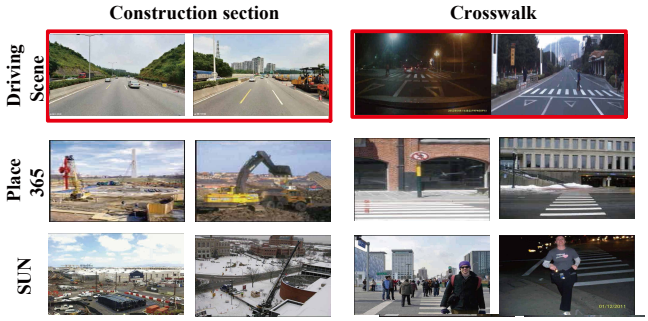


Fig. 7: Sample images of density evaluation. The most similar pair is highlighted in red.

A. Evaluation Metrics

Evaluation Metrics for Datasets The proposed dataset consists of about 0.1 million road scene images, in which about 60% are captured in real scenes and the rest is collected from the internet. A description of the categories has been given in Fig. 1. We measure the relative densities and diversities between SUN, Places365 and DrivingScene in 9 shared categories. The ground-truth measurements are obtained under the same protocol: a number of pairs of images are provided, and human annotators point out a pair with the highest similarity among all pairs. We observed that there is good consistent for different annotators in doing this task. We follow the settings in Places365 [10] in our experiments. As shown in Table III,

Dataset	Bridge	CommonRoad	Construction section	Crosswalk	GasStation	Highway	ParkingLot	Parking space area	Park Road	Avarage
SUN	0.600, 0.750	0.440, 0.750	0.480, 0.783	0.600, 0.683	0.440, 0.650	0.400, 0.750	0.180, 0.817	0.360, 0.833	0.180, 0.717	0.409, 0.748
Places	0.610, 0.667	0.500, 0.550	0.900, 0.633	0.460, 0.633	0.660, 0.650	0.880, 0.650	0.560, 0.733	0.360, 0.700	0.620, 0.633	**0.617**, 0.650
DrivingScene	0.260, 0.867	0.560, 0.85	0.320, 0.933	0.440, 0.800	0.400, 0.783	0.220, 0.817	0.760, 0.750	0.780, 0.767	0.700, 0.700	0.582, **0.807**

TABLE II: The diversity and density value of each subclass in dataset SUN, Places and DrivingScene.

specification	Number of trials	Number of pairs in each trial	Number of annotators
Diversity	40	12 pairs	2
Density	25	12 pairs	2

TABLE III: The experimental set and specification for quantifying density and diversity in each category.

the diversity and density in each category are examined with 40 and 25 trials, respectively. Each trial contains 12 pairs of images. For each pair, 2 annotators are employed to quantify the according value.

For the diversity experiment, the pairs are randomly sampled from each dataset. Each trial is composed of 4 pairs from each dataset, which results in a total of 12 pairs. Then the annotators select the most similar pair on each trial. 40 trials are performed on each of the 9 shared categories, and are independently judged by two annotators. The formula for calculating diversity of DrivingScene is as: $1-p(similar(pairs \in DrivingScene) < similar(pairs \in other\ datasets))$. Figure 6 shows some examples of image pairs from one of the diversity experiments. The pair selected by annotators is highlighted in red. Table II shows the average diversity over all categories for each dataset (the first number of each item), the average relative diversity is 0.58 for DrivingScene, which is higher than SUN's 0.41 and slightly lower than Places365's 0.62.

The density experiment is based on the visual similarity between images, which means the pairs how to generate is different from the diversity experiment. Intuitively, this would require firstly to find the most similar one of each image, which would be experimentally expensive if it is done by human. Instead, here we represent the visual similarity by the Euclidean distance between the Gist descriptor [44] of two images. We also select 1 frame from every 30 in the sequence to avoid heavy scene overlaps. Each image pair is composed of one randomly selected image and its 5-th nearest neighbor measured by Gist. In this experiment, we show 12 image pairs at each trial, but run 25 trials per category instead of 40 to avoid duplicate queries. Figure 7 shows some examples of image pairs in the density experiments and the selected image pair is also highlighted in red. Table II shows the average density over all categories for each dataset (the second number of each item), where the DrivingScene holds an average value of 0.807, the highest among the three datasets.

According to the density and diversity values in the above statistic analysis, the DrivingScene holds the same standards as the SUN and the Place.

Evaluation Metrics for Models Up to date, there are few methods proposed for effectively training network among

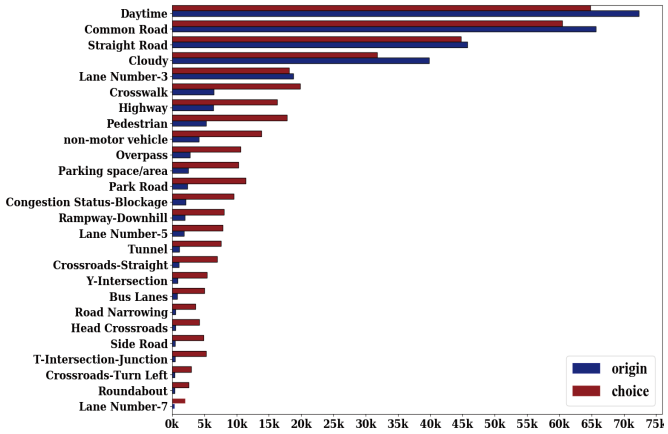


Fig. 8: Average image number within each class before and after applying the deep integration method. Images are adaptively resampled by AdaBoost algorithm.

Model	mAP (%)
VggNet	74.9
GoogleNet	76.5
ResNet-50	76.1
VggNet+Data Integration	81.0
GoogleNet+Data Integration	81.3
ResNet-50+Data Integration	81.1

TABLE IV: mAP value over all classes for different multi-label architectures.

imbalanced categories and most of them still transfer the multi-label task into training a single-label model. As no unified measurement is available to evaluate the quality of classifiers trained by imbalanced dataset, we follow the protocol of mean average precision (mAP) [45]. Precision and Recall metrics are usually considered because of imbalance of a dataset, we also show the PR-curve for three groups in Fig. 10.

B. Ablation Study of Proposed Approach

Ablation Study for Backbone Networks In this experiment, we evaluate the proposed method by using three different basic architectures: the VggNet [31], the GoogleNet [39] and the ResNet-50 [46]. The GoogleNet is deployed in our network with implementation details discussed in Section IV. Since the VggNet has a relative shorter length, we only set two loss layers. The first lies after the layer of $pool4$, whose feature map is filtered by an additional convolutional layer with a kernel of 1×1 pixels to reduce the feature dimension. The second loss layer is at the output which is the same as

in the GoogleNet. The corresponding weight decay values are equally set to 0.5. The ResNet shares a similar structure as the VggNet but with a much larger depth. Thus, we add three loss layers to it, which are the same as in the GoogleNet. For both nets, the weight decay values are respectively set to 0.2, 0.2 and 0.6. In the training, similar to [46], for all networks, the initial learning rate is 0.1 and the total number of iterations is 220k. In Table IV, we list the mAP values over all classes for the architectures in comparison, under different configurations. In general, the deeper structure would lead to better classification results. For example, Googlenet and ResNet-50 are both better than VggNet in classification. The best result is based on Googlenet, which is 76.5%.

Ablation Study for Data Integration Further, as the data integration is incorporated in our training process, the degree of imbalance among multiple classes can be significantly reduced. The scale of a few classes used for $L_{softmax}$ to boosting is illustrated by red bars in Fig. 8. We can notice that, not all samples of the majority class will be selected for low-level enhancement, although small classes have larger selection weight. The network is soft and stable, where the big classes will also have enough samples selected at the top level of the output side. In this way, the classification power can be enhanced for all classes in our approach. In average, a performance gain of more than 4.8% has been obtained in all backbone networks, as can be seen in Table IV.

Ablation Study for Resolution-adaptive Mechanism In the above, we introduced the multi-label classification network trained by cropping the input image size to 448×448 pixels, and we further fine turn network with the resolution-adaptive mechanism, the SPP [42] and ROI [41] approach. All the approaches are tested on two image groups, which are generated with respect to a resolution threshold of 2.8M pixels. We calculate the mAP value for each method over all the classes and the test results are reported in Table V. For the group of small images, the precision of our method is a little bit better than other compared methods, a performance gain of up to 1.5%, indicating that the resolution-adaptive mechanism is also applicable to small images. For images with big resolutions, the cropping operation may lead to loss of label property for inappropriately selected image regions, as the resolution-adaptive mechanism can adjust the crop size according to the image size, we achieve a gain of up to 3% in precision in comparison with other approaches. Since the area of pooling operation in small image will not be distorted like SPP or ROI, we achieve the highest precision of 84.2%. In Fig. 9, we illustrate the learned features after the resolution-adapted layer for several scenes. These features are represented by gray images with bright pixels to indicate weights of features. As shown, the dominant features are nearly from regions which are most relevant to the image labels.

C. Quantitative Analysis on the DrivingScene

In the above experiment, we gave a combination of all our methods, and each of our methods has achieved significant improvements over other common solutions to imbalances. We chose the following four models as our comparison method,

Method	Crop-448	ROI	SPP	Resolution-adaptive
Small images	83.6	84.3	84.6	**85.1**
Big images	79.0	81.9	82.0	**83.3**
All images	81.3	83.1	83.3	**84.2**

TABLE V: The mAP value over all classes for compared approaches in dealing with varied input image size.

Method	Instance	Place	Weather	Road Structure	All Class
Baseline1	65.1	74.9	81.2	68.6	71.9
Baseline2	65.6	75.0	84.2	71.7	73.5
Boosted Cascaded	75.8	76.2	88.7	76.1	76.5
Quintuplet Sampling	76.6	78.8	89.6	**79.0**	79.7
Our Approach	**77.1**	**79.3**	**91.5**	78.3	**81.3**

TABLE VI: mAP values tested on the top four super classes of our dataset in comparison between four baseline models and our approach.

all the compared methods exhibit an ability in dealing with the imbalance among multiple categories.

- Baseline1 resamples the foreground and background image patches for learning a convolutional neural network and achieves the best classification result according to the recent study of [34]. Follow this strategy, we estimate the attention area for our four super classes. For an image sample to be classified, the label belonging to the weather category is assigned to the top area of the image. The road structure class and the road instance class are usually found in the bottom area of images. And the place class is mostly related to the central image area. If more than one category appears in same image, the minority class will be prioritized.
- Baseline2 encodes super classes by the knowledge from a confusion matrix in [38], which proposes a multi-scale architecture with two CNNs. The shallow CNN is used to extract features of the super classes and aims to integrate minority class information while the deeper CNN takes high-resolution images as input to identify subcategories, the final output of Baseline2 can be obtained by the average of the shallow and the deeper CNN. This work can be treated as one of the cost-sensitive learning approaches by aggregating minority classes into majority classes.
- Baseline3 [36] combines multi-level resampling into the deep-learning structure. Here, we train the boosted cascaded convolutions [36] in 3 levels since the loss of the network is stabilized.
- Baseline4 uses a Quintuplet sampling strategy [37], with parameters unaltered. In the experiment, clusters for each class are formed with a size of $l = 200$ and a number of $k = 20$ nearest clusters are searched for querying. The clusters will be recalculated every 5,000 iterations.

Those above methods as well as our approach are tested on the DrivingScene dataset and the results are listed in Table VI. It can be observed from Table VI that, our model achieves a gain of 1.6% to 11.2% comparing with other methods. Table VII shows the mAP values on each sub-

Fig. 9: Test examples by the GoogleNet. For each image we display its groundtruth-labels. Gray images on their right side are feature maps generated by the resolution-adapted layer, with bright pixels to indicate greater weights.

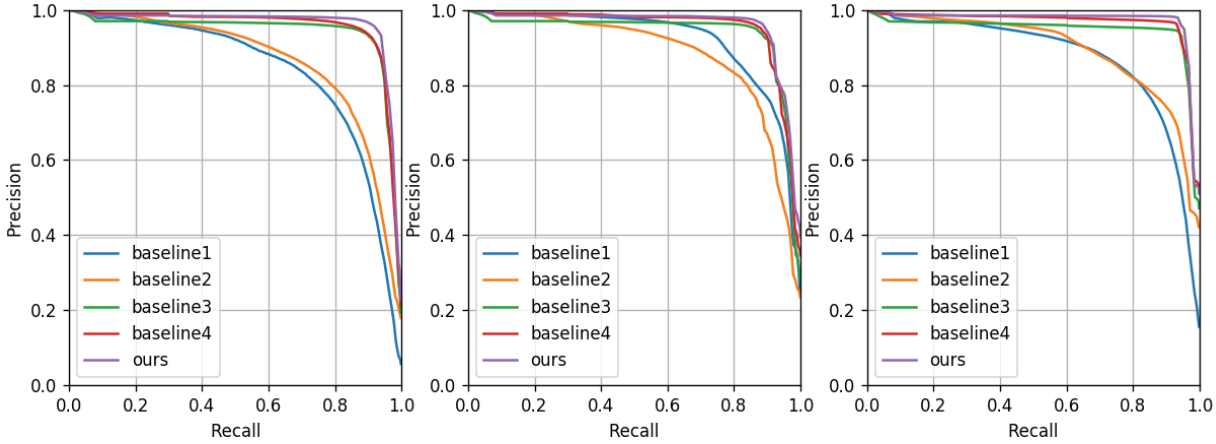


Fig. 10: PR-curves for three tag groups with comparison of different models. From left to right we respectively show PR-curves for small groups, middle groups, and big groups, as defined in Section IV-A.

category of DrivingScene. Compared with boosted cascaded convolution [36] w.r.t. precision values reported in Table VII, our model achieves a gain of about 1% to 4%, which is because the ensemble of multi-level results in network [36] has not changed the features learned only with class re-sampling.

From Table VII we can also see that, comparing with Quintuplet sampling [37], our model obtains higher performance on majority classes. The Quintuplet sampling maximizes the distance between majority and minority classes in the classification space to overcome data imbalance. Thus it demonstrates powerful feature extraction capabilities and is more robust for minority categories. However, the side effect of Quintuplet sampling is that the distance among majority classes is reduced. Therefore, it is inferior to our method on handling big classes, *e.g.*, with weather tags like Cloudy or place tags like Common road.

Figure 11 gives some example results to show the shortcomings of the five models. From Fig. 11 we can see, the recognition effect of the road attributes is relatively lower than that of other structures. This is because, the features in road attributes used for classification are not obvious enough, especially in the absence of clear lane markings, moving pictures, complicated intersections and visual field blockage. Overall, we can see that there is still a large room of improvement for each method.

In Section IV-A, 52 categories are sorted into three groups: small tag group cls_{small} with less than $1,000$ samples, medium tag group cls_{med} with a sample number between $1,000$ and $10,000$, and big tag group cls_{big} with more than $10,000$ samples. The class number in the three groups is 17, 22 and 13, respectively. The PR-curves have been plotted in Fig. 10. It can be seen from Fig. 10, our method has obvious advantages over two basic models while holding slightly better performance than the other two models. This observation is consistent with the previous analysis.

D. Results in PASCAL VOC 2012

The work [10] has shown that the learned higher-level features are different between object-centric and scene-centric CNNs, to provide more insights about the performance of our approach, we run test on object-centric databases, *i.e.*, the PASCAL VOC 2012 [47]. Baseline1 in [34] applied box information on samples of the PASCAL VOC dataset and is trained by transfer learning with the help of ImageNet. Thus, its mAP value reached 82.8% in [34]. However, this approach targets at learning and transferring mid-level image

Super class 1	**Traffic status: Smooth**	**Traffic status: Slow**	**Pedestrian**	**Non-motor vehicle**	**Waiting at red light**	**Construction section**	**Water covering road**
(Percentage)	(14.35%)	(2.78%)	(1.23%)	(0.96%)	(0.31%)	(0.23%)	(0.11%)
Baseline1	79.1	66.2	66.7	66.7	61.1	67.0	63.9
Baseline2	82.3	62.9	63.2	67.8	64.1	63.8	64.4
Boosted Cascaded	86.1	78.0	74.1	74.3	72.1	78.8	75.0
Quintuplet Sampling	84.7	77.9	75.3	74.9	72.2	**80.1**	**76.8**
Ours	**88.0**	**78.3**	**75.6**	**75.1**	**73.8**	79.6	75.3
Super class 2	**Common road**	**Highway**	**Park road**	**Toll station**	**Bus station**	**Tunnel**	**Parking lot**
(Percentage)	(15.28%)	(1.49%)	(0.55%)	(0.39%)	(0.31%)	(0.20%)	(0.11%)
BaseLine1	85.3	78.3	77.6	69.7	63.9	80.1	66.8
Baseline2	86.2	79.7	77.8	72.4	65.2	80.4	67.0
Boosted Cascaded	89.4	86.3	84.9	78.6	69.7	83.2	71.4
Quintuplet Sampling	86.9	86.1	85.3	79.2	**70.9**	85.3	**72.0**
Ours	**89.5**	**88.0**	**87.6**	**79.7**	70.1	**88.7**	71.8
Super class 3	**Daytime**	**Cloudy**	**Sunny**	**Night**	**Rainy**	**Dusk**	**Haze**
(Percentage)	(16.83%)	(9.27%)	(6.31%)	(0.43%)	(0.32%)	(0.29%)	(0.10%)
BaseLine1	86.7	90.7	74.3	87.4	74.8	83.9	70.8
Baseline2	88.6	91.1	78.9	89.7	78.4	89.3	73.8
Boosted Cascaded	88.1	93.4	86.7	87.4	78.9	95.7	80.8
Quintuplet Sampling	89.9	92.9	86.1	90.1	85.0	95.3	**87.8**
Ours	**95.6**	**95.0**	**87.3**	**91.2**	**89.0**	**95.5**	86.9
Super class 4	**Lane number: 3**	**Lane number: 2**	**Lane number: 1**	**Ramp: Downhill**	**Ramp: Uphill**	**Y-Intersec.: Turn Right**	**Y-Intersec.: Turn Left**
(Percentage)	(4.38%)	(3.89%)	(0.85%)	(0.46%)	(0.27%)	(0.20%)	(0.13%)
BaseLine1	73.9	81.9	72.1	71.2	72.7	60.7	62.7
Baseline2	78.1	77.9	81.8	72.0	70.7	61.2	64.6
Boosted Cascaded	75.8	73.4	72.1	71.1	72.3	63.1	65.0
Quintuplet Sampling	75.1	73.6	81.8	**73.7**	**74.1**	**66.9**	**67.1**
Ours	**76.0**	**78.5**	**75.0**	73.6	73.9	66.2	66.3

TABLE VII: mAP values on subcategories of each super class. For each subcategory, its percentage in the entire dataset is displayed under its name. Baseline1 [34] and Baseline2 [38] are two single-labeled models. Another two compared methods are the Boosted cascaded [36] and Quintuplet Sampling [37].

Class	**person**	**bird**	**cat**	**cow**	**dog**	**horse**	**sheep**	**plane**	**bicycle**	**boat**	**bus**	**car**	**motor**	**train**	**bottle**	**chair**	**table**	**plant**	**sofa**	**tv**	**Overall**
Baseline1	88.1	65.3	64.7	55.3	62.9	69.2	65.3	91.8	66.1	55.2	80.2	69.1	80.8	76.4	47.3	57.1	55.0	44.6	50.5	68.1	65.7
Baseline2	87.4	66.9	72.0	58.6	73.2	69.6	61.0	90.4	68.1	57.7	80.6	70.5	73.4	77.1	42.3	56.3	55.9	46.7	54.2	68.7	66.5
Boosted Cascaded	89.1	67.4	76.2	59.6	73.4	67.0	66.3	92.6	71.0	67.9	79.3	73.2	80.8	78.1	45.6	59.5	57.0	49.7	45.9	68.1	68.3
Quintuplet Sampling	89.3	67.8	79.8	54.5	67.0	**79.1**	70.3	90.1	76.3	**71.1**	83.5	74.2	79.1	78.9	**50.6**	**63.8**	**58.9**	50.7	60.4	68.7	70.7
Ours	**93.1**	**69.4**	**80.8**	**62.1**	**75.8**	78.8	**70.7**	**92.8**	**76.9**	66.8	**85.8**	**83.7**	**81.6**	**83.1**	49.8	62.7	58.3	**51.5**	**61.9**	**70.9**	**72.8**

TABLE VIII: mAP values for each category of Pascal VOC 2012. For each subcategory, its percentage in the entire dataset is also displayed under its name. Baseline1 [34] and Baseline2 [38] are two single-label models. The other two compared methods are the Boosted cascaded [36] and the Quintuplet Sampling [37]. The last column displays the mAP values over all categories.

representations using CNNs and the bounding box of object is the key for this high mAP value. To be fair, we don't use the box information for learning high-level features in our experiment and only use the ImageNet as the initialization parameter and sample center areas of the image with random offsets. Test results on Table VIII show that the precision gain achieved by our method, in comparison with the baselines, on the average, is over 2%.

E. Visualization of the Deep Features

In the above chapters we have demonstrated the effectiveness of our method. Here we would like to explore the leveraged image information by the network through visualization of utilized deep features. The technique of convolution visualization has been progressively developed in recent years and related researches can be roughly divided into two categories: the dataset-centric and the network-centric approach. The former one requires to train a DNN and afterwards to feed the data into the network; The latter one, however, only requires the trained network itself. Although the latter procedure is a relative simple, the former is generally accepted in most works because it has a more clear visual effect. In this experiment, we first utilize the test set of DrvingScene (*i.e.*, 33k images) as the input for the network. Then we sort all the images according to the activation responses of neural units in one layer. Finally we take the top 100 deconvolution images with the largest responses as the receptive field (RF) visualization of the units. This work is done with the visualization tool

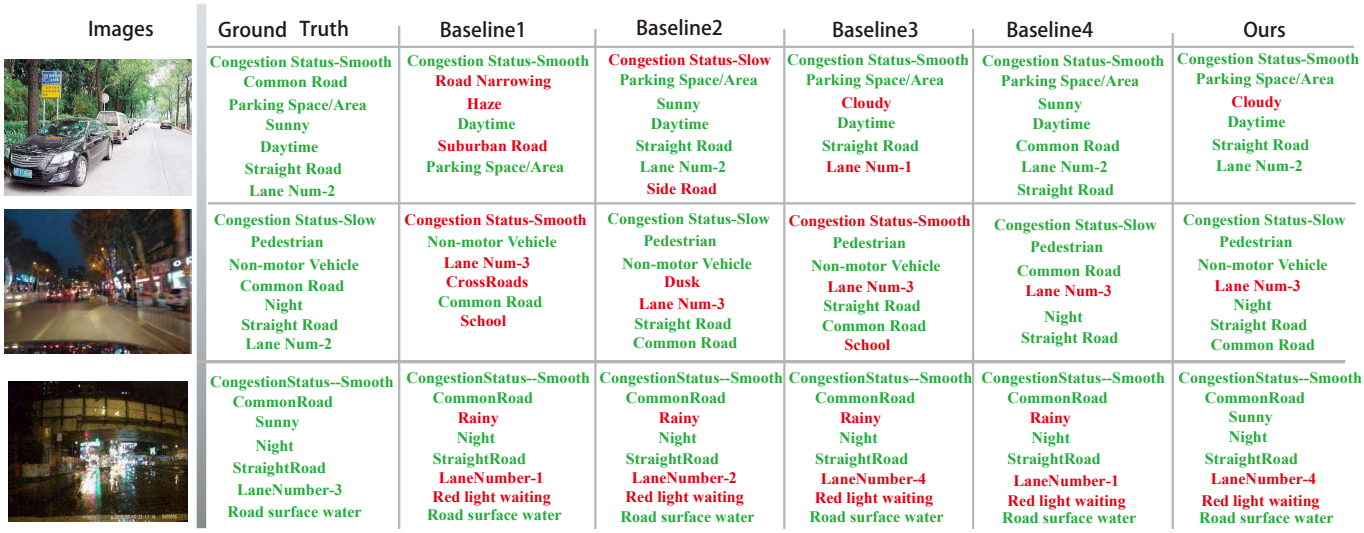


Fig. 11: Scene images and the corresponding ground-truth labels are displayed in the first two columns. The evaluation results of four baseline methods and our method are displayed in rest columns. Baseline1 [34] and Baseline2 [38] are two single-label models. The other two compared methods are the Boosted cascaded [36] and the Quintuplet Sampling [37]. In the results, the correctly predicted labels are denoted in green and the false predictions are denoted in red.

of [48].

Image samples of visualized receptive fields are illustrated in Fig. 12. For a better comparison, we also show images with maximum neural neural units activations, illustrated in Fig. 13. For each image, we also present the predicted labels (text in black areas under each single image in Fig. 13), which are ranked w.r.t. their scores. As can be seen, in the first row of Fig. 12, the purple areas at the bottom of each image represent the lanes or intersections to be distinguished. The bright white and light pink areas (*e.g.*, in images of the second row) indicate the most effective image regions for weather classification, lines of cross walks and lane markings are the most obvious parts in the image. The third row includes place information such as overpass and gas station, which are highlighted in receptive fields. In the last row, to infer the road trend, vehicle and pedestrians are also highlighted. Because the current pose and moving direction of the vehicle or pedestrian are usually consistent with the trend of the road, which can be considered as a powerful evidence for the classification procedure. Fig. 12 and Fig. 13 show that the learning features of CNN are consistent with our empirical judgement.

VI. Conclusion

In this work, we contribute with a large-scale dataset for the self-driving scene recognition, consisting of images mostly captured in real traffic scenarios and rich in both class density and diversity. Our DrivingScene, in contrast to many existing computer vision datasets it is: 1) imbalanced, because it was collected from different resources, 2) more representative of real-world road scene recognition challenges than previous datasets, 3) and suitable for investigating the multi-label scene classification problem. Based on the challenges of this dataset, we present a new network architecture incorporating hybrid-labels in multi-level loss functions and a deep data integration method to rebalance the class prior and enhance classification power on misclassified samples. By applying a resolution adaptive mechanism, we are capable to directly extract feature from different input image sizes, maintaining the most image information. Under the proposed mechanism, the dataset was proved to be effective in training the deep convolutional networks. The work of this paper would be of great interest to the autonomous-driving community, as regarding to the lack of such large driving-scene datasets and effective methods for multi-class classification under data imbalance.

Unlike traditional, researcher-collected datasets, the DrivingScene dataset has the opportunity to grow with the Self-driving community. Thus, the current challenges of the dataset will become more relevant. In the future we plan to investigate additional annotations such as more road scene attributes, location, variation environmental conditions, *etc*.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1305002 and the National Natural Science Foundation of China under Grant 61773414 and 61872277.

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1052–1067, 2007. 1

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016. 1

[3] Q. Li, L. Chen, M. Li, S. Shaw, and A. Nuchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 540–555, 2014. 1

Fig. 12: Deconvolution visualization of receptive fields for the units at the layer of *Pool5*. Image samples are from the DrivingScene dataset.

[4] D. Gonzlez, J. Prez, V. Milans, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016. 1

[5] L. Chen, L. Fan, G. Xie, K. Huang, and A. Nuchter, "Moving-object detection from consecutive stereo pairs using slanted plane smoothing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3093–3102, 2017. 1

[6] J. Wei, J. M. Snider, T. Gu, J. M. Dolan, and B. Litkouhi, "A behavioral planning framework for autonomous driving," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 458–464. 1

[7] L. Chen, X. Hu, T. Xu, H. Kuang, and Q. Li, "Turn signal detection during nighttime by cnn detector and perceptual hashing tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3303–3314, 2017. 1

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016. 1, 2, 4, 5

[9] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Feifei, "Imagenet: A large-scale hierarchical image database," *European Conference on Computer Vision*, pp. 248–255, 2009. 1, 2

[10] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018. 1, 2, 4, 5, 8, 11

[11] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. 1

[12] L. Chen, X. Hu, T. Xu, H. Kuang, and Q. Li, "Turn signal detection during nighttime by cnn detector and perceptual hashing tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3303–3314, 2017. 1

[13] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *European Conference Computer Vision (ECCV)*, 2016. 1

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *computer vision and pattern recognition*, pp. 3431–3440, 2015. 1

[15] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "Deepcrack: Learning hierarchical convolutional features for crack detection," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, 2019. 1

[16] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," *international conference on computer vision*, pp. 2758–2766, 2015. 1

[17] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," *computer vision and pattern recognition*, pp. 4040–4048, 2016. 1

[18] L. Chen, M. Cui, F. Zhang, B. Hu, and K. Huang, "High speed scene flow on embedded commercial-off-the-shelf systems," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2018. 1

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 1

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, 2015. 1

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015. 1

[22] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in

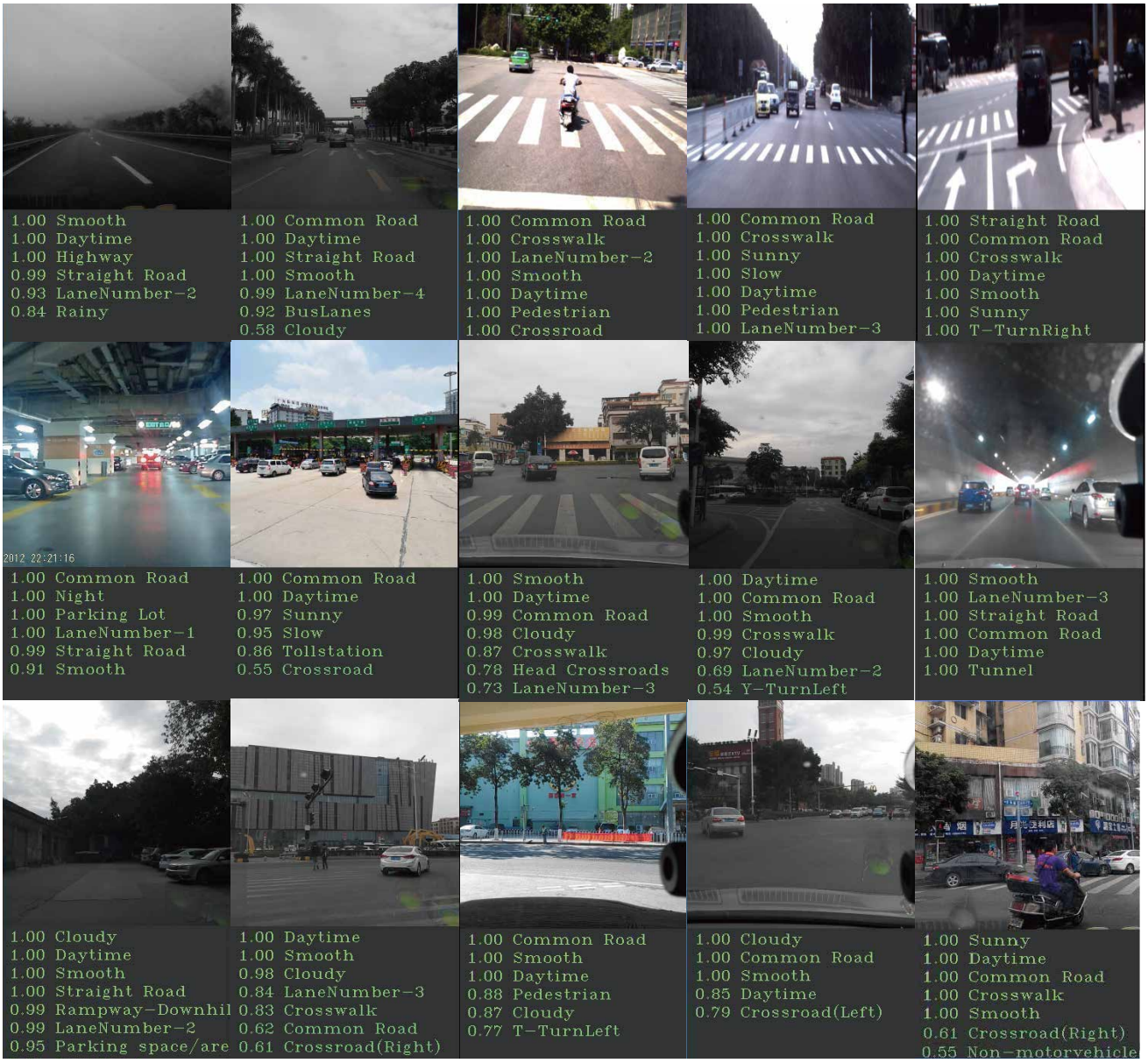


Fig. 13: Image samples with predicted labels by the VggNet. The scores and scene classes of each image is shown below.

context," *IEEE Conference on European Conference on Computer Vision*, pp. 740–755, 2014. 1, 2

[23] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010. 2, 4

[24] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012. 2, 4, 5

[25] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3973–3981, 2015. 2

[26] I. Sikirić, K. Brkić, J. Krapac, and S. Šegvić, "Image representations on a budget: Traffic scene classification in a resstricted bandwidth scenario," *IEEE Intelligent Vehicles Symposium*, 2014. 2

[27] Y. Luo, T. Liu, D. Tao, and C. Xu, "Multiview matrix completion for multilabel image classification," *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2355–2368, 2015. 2

[28] X. Li, X. Zhao, Z. Zhang, F. Wu, Y. Zhuang, J. Wang, and X. Li, "Joint multilabel classification with community-aware label graph learning," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 484–493, 2016. 2

[29] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2285–2294, 2016. 2

[30] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan, "Instance-aware hashing for multi-label image retrieval," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2469–2479, 2016. 2

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations*, 2015. 2, 3, 9

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016. 2, 3

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 2, 3

[34] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," *IEEE conference on computer vision and pattern recognition*, pp. 1717–1724, 2014. 2, 3, 5, 10, 11, 12, 13

[35] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," *IEEE conference on computer vision and*

pattern recognition, pp. 5375–5384, 2016. 2

[36] P. Kumar, M. Grewal, and M. M. Srivastava, "Boosted cascaded convnets for multilabel classification of thoracic diseases in chest radiographs," *international conference on image analysis and recognition*, 2017. 3, 10, 11, 12, 13

[37] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," *computer vision and pattern recognition*, 2016. 3, 10, 11, 12, 13

[38] L. Wang, S. Guo, W. Huang, Y. Xiong, and Y. Qiao, "Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns," *IEEE Transactions on Image Processing*, 2017. 3, 5, 10, 12, 13

[39] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv*, 2016. 5, 9

[40] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," *European Conference on Computational Learning Theory*, 1995. 6

[41] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. 7, 10

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015. 7, 8, 10

[43] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *international conference on learning representations*, 2014. 8

[44] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001. 9

[45] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, "Learning from noisy large-scale datasets with minimal supervision," *arXiv:1701.01619*, 2017. 9

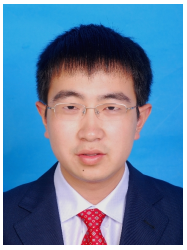
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. 9, 10

[47] M. Everingham, L. Van Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010. 11

[48] J. Yosinski, J. Clune, A. Nguyen, T. J. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv: Computer Vision and Pattern Recognition*, 2015. 13

Long Chen received the B.Sc. degree in communication engineering and the Ph.D. degree in signal and information processing from Wuhan University, Wuhan, China, in 2007 and in 2013, respectively. From October 2010 to November 2012, he was co-trained PhD Student at National University of Singapore. He is currently an Associate Professor with School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He recieved the IEEE Vehicular Technology Society 2018 Best Land Transportation Paper Award, IEEE Intelligent Vehicle Symposium 2018 Best Student Paper Award and Best Workshop Paper Award. His areas of interest include autonomous driving, robotics, artificial intelligence where he has contributed more than 50 publications. He serves as an Associate Editor for IEEE Technical Committee on Cyber-Physical Systems newsletter, and Guest Editor for IEEE Transactions on Intelligent Vehicles.

Wujing Zhan is with School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, P.R.China. Now he is a postgraduate student and studies machine learning and deep learning in applications of scene recognition, scene segmentation and scene flow.



Wei Tian received the B.Sc degree in mechatronics engineering from Tongji University, Shanghai, China, in 2010. From October 2010, he was with the Department of Electrical Engineering and Information Technology at KIT, Karlsruhe, Germany, and received the M.Sc. degree in May 2013. He is currently working toward the Ph.D. degree at the Institute of Measurement and Control Systems at KIT. He is interested in research areas of robust object detection and tracking.

Yuhang He received the B.Sc degree in school of Geodesy and Geomatics from Wuhan University, Wuhan, China, in 2013. He is currently working on fashion related topics and autonomous driving. His research interest spans from computer vision, machine learning to robotics.

Qin Zou received his B.E. degree in information science and Ph.D. degree in photogrammetry and remote sensing (computer vision) from Wuhan University, Wuhan, China, in 2004 and 2012, respectively. From 2010 to 2011, he was a visiting PhD student at the Computer Vision Lab, University of South Carolina, USA. Currently, he is an Associate Professor with the School of Computer Science, Wuhan University. He is a co-recipient of the National Technology Invention Award of China in 2015. His research activities involve computer vision, pattern recognition, and machine learning. He is a member of the IEEE, and a member of the ACM.