doi:10.1109/BigData.2018.8622318
https://ieeexplore.ieee.org/document/8622318

Region-based Cycle-Consistent Data Augmentation for Object Detection

Florian Kluger*, Christoph Reinders*, Kevin Raetz†, Philipp Schelske†,
Bastian Wandt*, Hanno Ackermann* and Bodo Rosenhahn*
* *Institute for Information Processing,*
Leibniz University Hanover, 30167 Hanover, Germany
† *Ignaris UG (haftungsbeschraenkt), 30167 Hanover, Germany*

***Abstract*—Roads constitute a major part of the lives of everybody. Heavy use, for instance by cars and especially trucks, and even soil movement lead to visible damages. While major roads are regularly inspected, smaller roads often lack attention. It is therefore of great interest to have camera-based systems which can automatically detect and even classify damages.**

This report presents a system developed by the authors as part of the Road Damage Detection and Classification Challenge at the 2018 IEEE Big Data Cup [1]. Further contributions made here are techniques to augment the small set of training data. As a major contribution we also propose refinements to the dataset and evaluation metric to improve the challenge.

1. Introduction

Detecting damages on road surfaces is a useful application for road users and administrators alike. Less important roads are less frequently inspected, thus damages can escape attention. Since manual inspection by experts is expensive, it is of great interest to have an automatic system available which does not require sophisticated and expensive capture devices. This report presents a system to detect damages on road surfaces. It uses image data captured by cameras of mobile phones.

In addition to presenting results of the Road Damage Detection and Classification Challenge at the 2018 IEEE Big Data Cup [1], we will propose two improvements to the data. The first aims at increasing the limited amount of training data. We will explain how more data can be augmented by modern machine learning techniques using only the provided data. Furthermore, we argue that several of the ground truth labels are inherently ambiguous. For instance, classes whose definitions are based on particular orientations of cracks w.r.t. the camera cannot be easily discriminated even by a human expert. Here, the difference between two classes rests upon a rotation by $90°$. However, cracks in roads are often not only vertically or horizontally oriented but in-between. Please notice that this definition even depends on the orientation of the camera which takes the images. Moreover, there are several annotated damages that cannot be exactly determined to belong to exactly one class.

Starting from our experimental results, where two of the proposed methods achieve a Top-10 score in the challenge, we present an analysis of the results which leads to a proposal of several improvements of the dataset as well as the evaluation criteria. We present a simple yet effective guided labeling approach based on our inferred bounding boxes that significantly reduces labeling time and cost. Additionally, we propose to use a modified evaluation metric which does not penalize ambiguous annotation possibilities too heavily. Fig. 12 shows an example where both instances of the bounding box are reasonable but have a very low IoU. By setting a soft threshold these get a higher similarity score.

Summarizing, our contributions are:

- Provably competitive performance in the Road Damage Detection and Classification Challenge at the 2018 IEEE Big Data Cup.
- A data augmentation method using only the provided dataset.
- A guided labeling approach for quick annotation of incompletely labeled data.
- An evaluation metric that tackles the problem of ambiguous bounding boxes.

2. Dataset

For the *Road Damage Detection and Classification Challenge* within the *2018 IEEE Big Data Cup* a dataset consisting of around 9,053 images has been provided [1]. The images were recorded with a mobile phone which was attached to the dashboard of a car. The dataset consists of 80% training data and 20% test data. For each image of the dataset the position and type of road damages are annotated as rectangular bounding boxes. Overall nine different classes of road damages are annotated (D00, D01, D10, D11, D20, D30, D40). Because class D30 has only a few labels, it is ignored according to the challenge organizers.

3. Detection Methods

In this section we present different methods for object detection to tackle the problem of detecting and classifying road damages. We start by applying state-of-the-art objection detection methods, namely Faster R-CNN [2], RetinaNet [3] and a combination of random forests with neural networks [4]. We briefly describe the most important parts

of these approaches in the following sections. For detailed explanations we would like to refer to the respective papers. To combine the advantages of the different approaches we present an ensemble learning method in the last section. The evaluation on the test set of the Road Damage Detection and Classification Challenge is shown in Table 1.

3.1. Faster R-CNN

Faster R-CNN [2] is an object detection framework composed of two stages: a region proposal network (RPN) followed by an object classifier. In the first stage, a fully convolutional network (FCN) – based on e.g. a VGG network [5] or ResNet [6] – generates a feature map for the complete image. Operating on this feature map, a second network then predicts for each location in the image whether one of k different predefined anchor bounding boxes contains an object at said location, and computes refined bounding box coordinates. The second stage applies non-maximum suppression to the detected bounding boxes and feeds their corresponding features from the FCN feature map into a classification network, which predicts the classes of the detected objects.

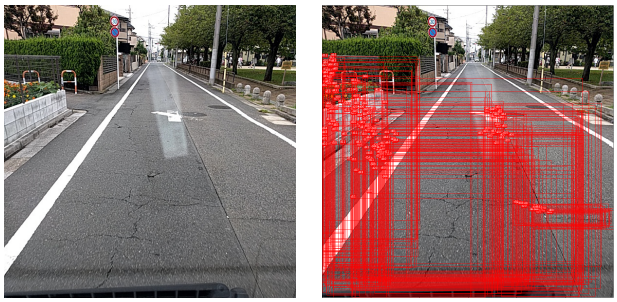
For our experiments we use a ResNet-101 as a base for the RPN and employ nine different anchors with sizes of 128^2, 256^2 and 512^2 pixels, and aspect ratios of $2:1$, $1:1$, and $1:2$.

3.2. RetinaNet

RetinaNet [3] is a state-of-the-art one-stage object detector with similar performance as two-stage detectors, for instance Faster R-CNN. It is composed of a feature pyramid network (FPN) on top of a feedforward ResNet architecture which computes a convolutional feature map over the entire input image, and two task-specific subnetworks. One subnetwork performs object classification on the output of the FPN, while the other performs bounding box regression. The FPN efficiently constructs a multi-scale feature pyramid of which each layer can be used for detecting objects of different scale. For each scale anchor boxes are generated which are then fed to the classification and regression subnets. The classification subnet predicts the potential class of an anchor and the regression subnet modifies the size of the original anchor to fit potential objects better. To counteract the imbalance between foreground and background classes that one-stage object detectors encounter during training, the RetinaNet replaces the standard cross entropy loss with the so called focal loss that down-weights the loss assigned to well-classified examples, thus focusing on harder examples. For our experiments we use a ResNet-50 as base network and anchor ratios of 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75 and 2.

3.3. Random Forest

End-to-end learning for object detection with convolutional neural networks is very successful and provides good results, as demonstrated by [2], [3]. Furthermore, convolutional layers in convolutional neural networks have shown to learn good feature representations which can be combined with other methods such as random forests [4]. Random forests are able to learn with small amounts of data and are very robust to overfitting.



(a) (b)

Figure 1: Based on a trained Faster R-CNN, we generate regional proposals. For better visualization only 30 region proposals per class are shown.

Based on the trained Faster R-CNN presented in Section 3.1 we use the region proposal network which is part of the object detection network to generate regional proposals. An example is shown in Fig. 1. For each image 300 region proposals are generated per class. In the next step, image features are generated by passing an image through the network and extracting features from the last convolutional layer before the object detection specific layers. The resulting image features in our network have a dimension of $38 \times 38 \times 1024$ consisting of 1024 filter outputs with a width and height of 38. To generate a feature representation for each region proposal we extract a patch of the image features depending on the position of the bounding box. Finally, we match each region proposal with the ground truth data and assign a label if the intersection over union (IoU) of a ground truth bounding box is greater than or equal to 0.5. Otherwise the region proposal gets the label *background*.

Our generated dataset has over 14 million training region proposals. By extracting the features and determining the labels, we transferred the problem to a classification task. Thus, we can train a random forest on the dataset. The dataset, however, is very large and does not fit into the memory. Therefore, standard methods cannot be applied. To circumvent that we learn each tree individually by selecting a random subset of region proposals, loading the features as well as the corresponding labels and train the tree on the subset of the data. In our application, we train a random forest with 100 individual trees and repeat the process for each tree.

During inference, the region proposals are extracted in the same way. Each region proposal is then classified using the trained random forest. Afterwards all region proposals with a probability higher than 0.1 are selected. Finally, we apply non-maximum suppression by iteratively selecting the

(a) (b)

Figure 2: Example for classifying region proposals with random forests. In (a) all region proposals with a probability higher than a threshold are shown and in (b) all filtered region proposals. The ground truth bounding boxes are shown in green.

Method	F_1
Faster R-CNN (Section 3.1)	0.610
RetinaNet (Section 3.2)	0.456
Random Forest (Section 3.3)	0.540
Ensemble Learning (Section 3.4)	0.602

TABLE 1: Evaluation on test set of the road damage detection benchmark. For each method the F_1 score is calculated.

bounding box with the maximum score and removing all other bounding boxes which overlap with an IoU greater or equal than 0.5. An example for classifying region proposals with random forests is shown in Fig. 2.

Overall, random forests combined with convolutional neural networks for feature generation achieve good results. In comparison to standard architectures based on convolutional neural networks, random forests are able to learn with very few training data.

3.4. Ensemble Learning

Looking at the results of our different detection methods we noticed that some methods outperform others on different target classes. Therefore, we aggregated the results of each detector for each image and sorted them by the confidence score of each network. Comparing all possible combinations the ensemble of the Faster R-CNN and the RetinaNet performs better than RetinaNet alone, but slightly worse than Faster R-CNN. Additionally, we filtered out predictions that had more than a 0.5, 0.6, 0.7, 0.8 or 0.9 IoU with a more confident prediction, but observed a decrease in overall detection score using this approach.

4. Findings

All state-of-the-art region detection approaches – even as an ensemble – created unsatisfactory results with a maximum F_1 score of 0.61. However, comparing to other submissions on the leaderboard we still are in the Top-10. In fact, the best 10 submissions only differ by an F_1 score

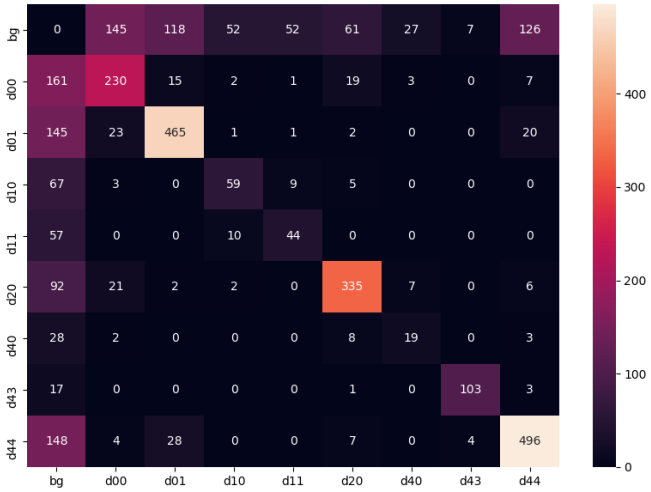
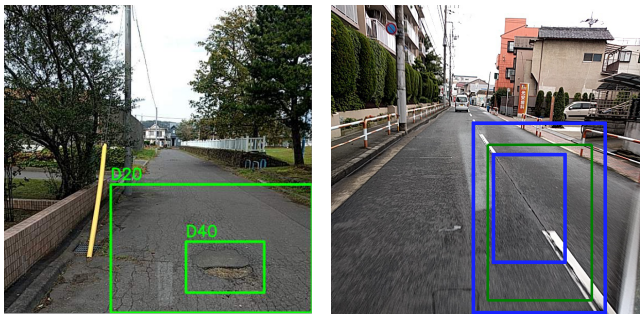


Figure 3: Confusion matrix for all damage classes. Every row corresponds to a detected class and every column contains the ground truth label. *bg* is the background class, i.e. an undamaged road or other objects.



(a) (b)

Figure 4: (a) Some damages are difficult to annotate, e.g. when different classes overlap. (b) Rectangular bounding boxes may be a suboptimal choice for some classes, as small ambiguities lead to large variances in IoU.

of approximately 0.1. This gave rise to the idea that there are some inherent flaws in the dataset itself. Taking an in-depth look at the outputs of our methods reveals three main factors for these low scores.

1) Due to overlapping classes of damages that are hard to detect (even by a human) the problem is indeed very challenging.
2) Rectangular bounding boxes are not optimal for long diagonal cracks.
3) Many clearly identifiable damages are not labeled in the ground truth data of the test set.

Figs. 4-5 show some examples for points 1) – 3). Since it is not possible for us to manually label all images by ourselves we calculated a lower bound for our method. We selected a random subset of 1,446 images from the training set and used it for validation purposes. Evaluating

(a) (b) (c) (d)

Figure 5: Examples of detections wrongly counted as false positives due to incomplete ground truth labels.

	bg	d00	d01	d10	d11	d20	d40	d43	d44
bg	0	35	29	15	5	46	15	3	51

TABLE 2: First column of the confusion matrix after relabeling. There is a significant difference compared to the original confusion matrix in Fig. 3

our detection algorithm on this validation set, we get the confusion matrix in Fig. 3. Every row corresponds to a detected class and every column contains the ground truth label. *bg* is the background class, i.e. an undamaged road or other objects. To define the lower bound we relabeled the images of the first column. These images contain damages detected by our network but marked as undamaged road. Every damage that is correctly detected in type and size is deleted from the confusion matrix. This gives a first column as in Table 2. These numbers are significantly lower than the original. If we ignore the class label and evaluate only for the presence of a damage in the image we have an average over all our labeling experts of 96.6% mislabeled images in the validation set. In total numbers it means that only 20 images from the first column of Fig. 3 are misdetections of our network. According to this findings we propose in the following several methods and new evaluation criteria to further improve the dataset.

5. Proposals

Our evaluation in Sec. 4 revealed some major flaws in the dataset. In the following we propose several approaches to improve the dataset, augment images of damaged roads for more synthetic training data, and more meaningful evaluation criteria. These proposals will significantly enhance the dataset as well as improve the challenge.

5.1. Generating new labeled data

Convolutional neural networks and other machine learning approaches achieve very good results in a wide range of applications. Such supervised learning methods however require labeled data. Especially convolutional neural networks are trained with huge amount of data. Generating labeled data is time consuming and very expensive.

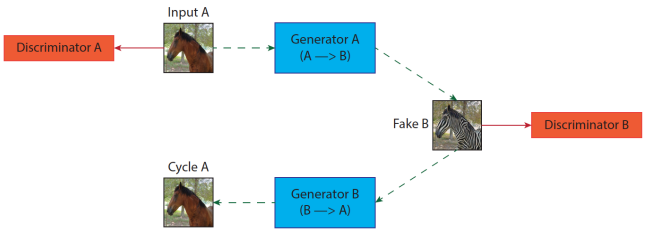
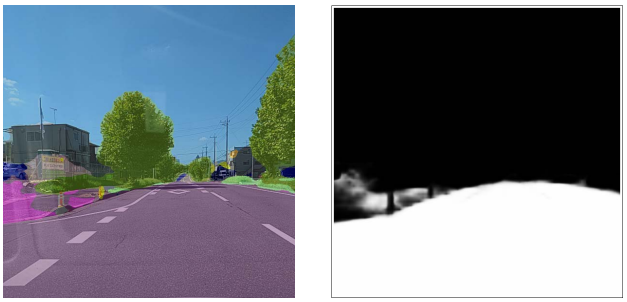


Figure 6: Schematic structure of a CycleGAN. Generator A gets an image from domain A as input and generates a fake image in domain B. Generator B does the same the other way around. The discriminators detect whether an image is real or fake in the respective domain.

Thus, we propose a pipeline for generating new road damages to increase the size and variety of the dataset based on Generative Adversarial Networks and Cycle-Consistent Generative Adversarial Networks, respectively. Generative Adversarial Networks (GAN) have been introduced by Goodfellow *et al.* [7] in 2014. Generative Adversarial Networks are based on a game-theoretic approach and consist of two networks - a *generator* and a *discriminator* - which compete against each other. The generator has the goal to generate real-looking images. The discriminator on the other hand should detect whether an image is real or fake. During the training, the two networks are in conflict with each other.

Cycle-Consistent Adversarial Networks (CycleGAN) are an extension of Generative Adversarial Networks developed by Zhu *et al.* [8] in 2017. The goal is to transform images from domain A to domain B without having paired images from domain A and B. In a CycleGAN two GANs are combined together where the input of the second network receives the output of the first network. In Fig. 6 a schematic structure of a CycleGAN is shown. Distriminator A decides whether an image from domain A is real or fake. Discriminator B does the same for domain B. Generator A gets images from domain A as input and generates images in domain B. Generator B does the same vice versa. The resulting cycle enable to transform images from domain A to domain B and back into the original domain A. The generated image should be equal to the original input image.



(a) (b)

Figure 7: The images are analyzed using a semantic segmentation (a) to detect the road (b) where potentially new damages can be generated.

CycleGANs work in both directions which means that the starting point can be domain A but also domain B.

5.1.1. Unpaired CycleGAN Dataset. To generate road damages, we define domain A as images without road damages and domain B as images with road damages. As described previously in Section 2 multiple classes are defined so that domain B is further divided into multiple subdomains based on the class of the road damages. For each subdomain in domain B an individual CycleGAN is trained. The images for domain B (images with road damages) are given by the annotations in the dataset.

Collecting images for domain A is more difficult because regions of the road without damages have to be found. To achieve this, we perform a semantic segmentation which classifies an image pixel-wise. We train a Pyramid Scene Parsing Network (PSPNet) [9] on the Cityscapes dataset [10] which has multiple classes including a road class. Afterwards we analyze each image from the road damage detection dataset using the trained PSPNet as shown in Fig. 7 and get the probability for each pixel that the corresponding pixel belongs to the road which is visualized in Fig. 7b. Additionally, we integrate the information of existing annotated road damages in the images and are able to generated regions of the road which do not have a road damage. For followup works, this could be further improved by operating on images rectified w.r.t. the ground plane, which can be computed automatically using vanishing points or the horizon line [11], as this would reduce variance in appearance due to the camera projection.

5.1.2. Generating Road Damages. CycleGANs are trained in two directions. On the on hand starting with domain A which is visualized in Figure 8. The input image is an image without a road damage and the generator adds a road damage to the image. This image is then used as input for generator B to remove the road damage again which completes the cycle. The image which is transformed back to domain A should be equal to the input image. The discriminators A and B try to detect whether an image is real or fake. During

(a) Domain A: real (b) Domain B: fake (c) Domain A: cycle

Figure 8: Example for Cycle A of the CycleGAN. Input image (a) without a damage is transformed into an image with road damage (b) which is than removed again (c).

(a) Domain B: real (b) Domain A: fake (c) Domain B: cycle

Figure 9: Example for Cycle B of the CycleGAN. Input image (a) with a damage is transformed into an image without a damage (b) which is than added again (c).

training the cycle is also performed the other way around starting with domain B as visualized in Fig. 9.

We trained a CycleGAN for each class where generator A generates road damages on images and Generator B removes road damages from the images. To generate new road damages we extracted possible regions as described in the previous section and generate road damage using generator A. As a result, new data can be created very easily to extend the dataset. Some examples are shown in Fig. 10. The left image show the original images and the right image the generated image where the new road damage is highlighted in yellow.

5.2. Modified Evaluation Metric

For evaluation during the challenge, the mean F_1 score is used as defined by:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{1}$$

with

$$\text{precision} = \frac{t_p}{t_p + f_p}, \quad \text{recall} = \frac{t_p}{t_p + f_n} \tag{2}$$

and t_p, f_p, f_n being the number of true positives, false positives and false negatives respectively. A detected bounding box is counted as a true positive if its IoU with a ground truth bounding box of the same class is greater than 0.5. Due to the nature of the problem and characteristics of the ground truth labels, this metric appears to penalize

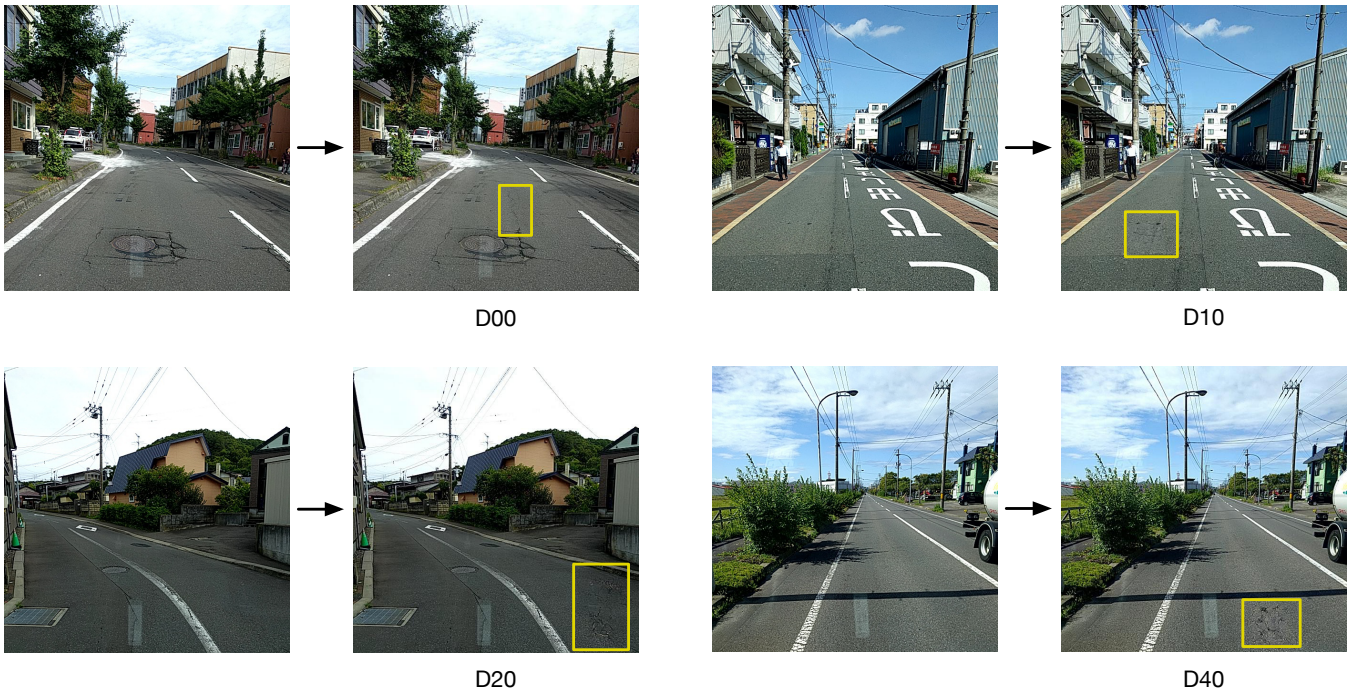


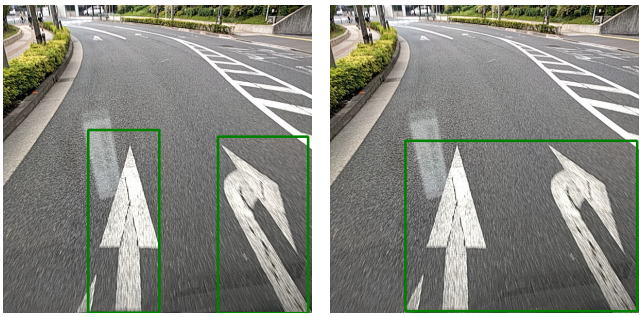
Figure 10: Generating new road damages for training in areas without a damage. Each image pair shows the original image on the left side and the image with a generated road damage (highlighted in yellow) on the right side. Existing bounding boxes are not shown.

reasonable yet imperfect (w.r.t. the ground truth) detections to a great extent, as mentioned in Sec. 4

Ambiguity w.r.t. the degree of segmentation: Unlike typical object detection problems based on bounding boxes where different instances are easily distinguishable, e.g. multiple chairs, cars, people etc. next to each other, it can be difficult to define where one damaged area of a road ends and another one starts. As illustrated by Figs. 11 and 12b, this can lead to a number of different bounding box configurations that appear plausible but are incompatible with each other w.r.t. the evaluation metric defined above.

Ambiguity w.r.t. bounding box size: For damages such as longitudinal cracks especially, it may not be clear where to define the start and end points of the damage. As illustrated by Figs. 4b and 12a, this ambiguity can lead to large variations in bounding box size and position.

We therefore propose to use a less discriminative evaluation metric without a firm threshold, for example the mean average precision (mAP) over a range of IoU values between 0.05 and 0.95 which is also used for the MS COCO [12] dataset. As exact localization of road damages appears of relatively little importance, a smaller upper bound for the IoU may be reasonable for this challenge.



(a) Our detection (b) Ground truth

Figure 11: Our method detected two bounding boxes (a) of the correct class, while the ground truth (b) combines them to a single large bounding box.

5.3. Guided Labeling

Manually labeling a large amount of data is not only time and cost intensive but also error-prone. To reduce costs and time for the dataset of the challenge we propose a guided labeling. In the current state there are some damages that have not yet been labeled (cf. Sec. 4). To improve the dataset without much manual re-labelling, we propose an automatic generation of region proposals by the already trained object detectors. Such region proposals can be easily checked without great time consumption. As long as the bounding box proposed by the damage detectors is correct (which it is in most cases) the new label can be easily set or rejected by the click of a button. In the case of bounding boxes of incorrect sizes, they can be quickly corrected. This procedure is far less time-consuming.

(a) (b)

Figure 12: Examples where the ground truth (green) is ambiguous, and reasonable alternatives (blue) with low IoU are possible.

5.4. Adjusted Classes via Clustering

As we noticed, some classes being quite similar can result in misclassifications. Since the reasoning behind the discrimination between some classes is not well defined, we suggest to first generate region proposals with class annotations for all damages and afterwards use a clustering algorithm, for instance K-means, mean shift or autoencoder based clustering algorithms, to generate clusters of all these region proposals. It then can be checked by experts to assign more meaningful class labels to them. This should result in more natural class labels, by merging similar classes together.

6. Conclusion

In this report we propose an ensemble of several state-of-the-art region detection methods well suited for the detection of road damages from mobile phone cameras. We achieved Top-10 results in the Road Damage Detection and Classification Challenge at the 2018 IEEE Big Data Cup. By deeply analyzing our results we found some formerly unknown and unexpected flaws in the dataset. This led us to propose a set of improvements for the dataset, the labeling process, and new evaluation metrics.

First, we propose a pipeline for artificially extending the dataset by generating new training data based on Cycle-Consistent Generative Adversarial Networks. Our CycleGAN is able to generate or remove a road damage in any specified region of an image. Second, we suggest new evaluation metrics in particular due to the fact that annotations are very difficult to define and may be ambiguous. Furthermore, we propose to employ a guided labeling to quickly annotate unknown or incompletely labeled images. By suggesting bounding box candidates to the user these can be efficiently accepted or modified by the click of a button.

In summary, we showed Top-10 performance for two of our approaches in the Road Damage Detection and Classification Challenge at the 2018 IEEE Big Data Cup. We are convinced that further challenges will benefit from our proposed methods for improving the dataset as well as the evaluation.

References

[1] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2017.324

[4] C. Reinders, H. Ackermann, M. Y. Yang, and B. Rosenhahn, "Object recognition from very few training examples for enhancing bicycle maps," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] 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," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] F. Kluger, H. Ackermann, M. Y. Yang, and B. Rosenhahn, "Deep learning for vanishing point detection using an inverse gnomonic projection," in *German Conference on Pattern Recognition*. Springer, 2017, pp. 17–28.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.