How to Improve Deep Learning based Pavement Distress Detection while Minimizing Human Effort

Daniel Seichter, Markus Eisenbach, Ronny Stricker, Horst-Michael Gross

***Abstract*— Aging public roads need frequent inspections in order to guarantee their permanent availability. Using deep neural networks, the process of detecting pavement distress can be automated to a high degree. However, evaluations show that they perform relatively poor on road images, that are significantly different from training data. Therefore, we show, how the performance can be improved with a human in the loop. The basic idea is to enlarge the training dataset. Luckily, many unlabeled road images from previous inspections are available. Nevertheless, annotating all of them is labor-intensive, and thus, not feasible. Since only diverse data enable an increase in performance, selecting the right subregions of the images for annotation is the key. To achieve this goal, we model the network's uncertainty and incorporate it for selecting new subregions. Our experiments show that we are able to improve the network's performance with only a fraction of data that would usually be necessary to get the same performance.**

I. Introduction

Public infrastructures suffer from aging and therefore need frequent inspection. Road condition acquisition and assessment are the keys to guarantee their permanent availability. In order to maintain a country's road network, millions of high-resolution images have to be analyzed annually. Currently, this requires cost and time excessive manual labor. Therefore, the time span between the actual inspection and the final evaluation may be up to several months. In the meantime, small damage, like cracks, can lead to substantial downtimes with a high impact on the population.

In previous work [1], [2], as part of the ASINVOS[1] project, we started to automate this visual inspection process to a high degree by applying deep neural networks for distress detection. The basic idea is to train a self-learning system with manually annotated data from previous inspections, such that the system learns to recognize underlying patterns of distress. Once the system is able to robustly identify intact infrastructure, it can reduce the human amount of work by presenting only distress candidates to the operator. This helps to significantly speed up the inspection process and simultaneously reduces costs. Furthermore, inspection intervals can be shortened, which helps to remedy deficiencies in time.

The evaluation in [1] shows, that the developed ASINVOS net (see Sec. III-A, Fig. 4) performs very well in segmenting road images, such that distress can be discovered easily. Nevertheless, the evaluations also showed that the performance drops on road images that are significantly different from training data. Therefore, the aim of this paper is to show, how the performance can be improved by a human in the loop while keeping the amount of manual labor low.

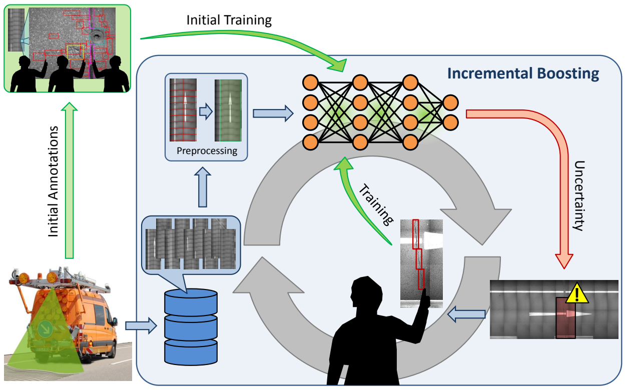

Fig. 1. Incremental boosting of distress detection performance. During inspections, lots of data are collected, that could help to improve the performance of a neural network. Since annotating these high-resolution images is very time consuming and labor-intensive, the neural network should decide by itself, which data would help to improve its performance. Therefore, a human operator has to annotate only small amounts of data. If the network has learned to handle these data, it will continue selecting different images for annotation, that currently lead to uncertain decisions.

For deep neural networks, we know that more diverse training data will improve their performance. Therefore, it might be a good idea, to train on a lot of data. Luckily, we can get many high-resolution road images from previous inspections. Each image shows ten meters of a lane. Based on German federal regulations, all high-resolution road images are coarsely annotated as intact or defect. The requested labeling is very coarse, in order to reduce the extent of manual labor. Every meter of the lane is divided into three parts of equal width (left, middle, right), resulting in a 10×3 grid for the whole image (Fig. 2). For each part not containing intact road, it is annotated which of five distress types (Fig. 3) is present in that subregion of the image. This means, the annotated subregion is ca. 1×1 meters large, even if a small crack within is only five centimeters long. Additionally, many annotations are spatially inaccurate or missing. Thus, this labeling is not sufficient for training or improving a neural network, that should segment the road image in detail (see [1]). Therefore, in order to be useful for training, each image needs to be annotated in detail. Annotating high-resolution road images, however, is very labor-intensive. Additionally, annotating many images

All Authors are with Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, 98694 Ilmenau, Germany. daniel.seichter@tu-ilmenau.de

This work has received funding from the German Federal Ministry of Education and Research as part of the ASINVOS project under grant agreement no. 01IS15036.

[1] ASINVOS – ASsisting and INteractive machine learning based Visual mOnitoring System for pavement surface analysis of roads and pipelines.

IEEE Int. Conf. on Automation Science and Engineering (CASE), Munich, pp. 63-68, IEEE 2018

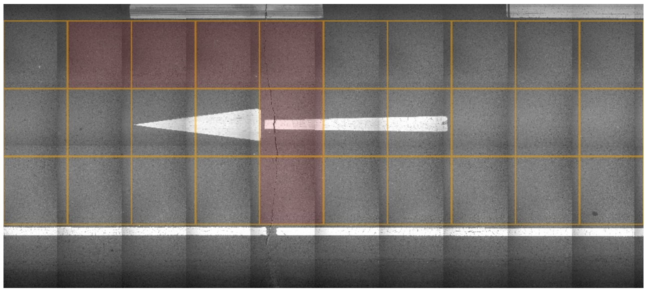

Fig. 2. Labeled high-resolution road image (7883×3498 pixels) as defined by German federal regulations. The label grid is highlighted in orange, and distress in terms of cracks is overlayed in light red. Even if a crack is very small, like the one on top, not even visible in this figure, the entire grid cell is labeled as distress. Thus, the labeling cannot be used to train a neural network for detailed segmentation.

is error-prone due to exhaustion. Thus, the strategy for fast improvement should be annotating only data that are diverse, instead of annotating as much as possible.

In our setting, deciding which images to label is crucial (Fig. 1). We need to evaluate when the neural network is uncertain in its decision. Therefore, it is important to know what the neural network does not know. We will show, how to achieve this requirement. Given a neural network that can decide which images are worth annotating, and even more, which parts of it, the human in the loop can concentrate on annotating these subregions in order to help the neural network to improve. If the neural network has learned to adapt to the new data, it will tell the operator that it is certain now. Hence, we do not need to annotate more data of this kind. In the next iteration, the neural network will focus on different data.

In summary, the contributions of this paper are as follows:

- We show how to train a neural network for distress detection such that we can determine what it does not know.
- We show how to select data that are worth annotating.
- We show that labeling only diverse parts of the data can be sufficient to improve a neural network as much as if it was trained on all data available.

II. Related Work

In this paper, we show how deep learning based pavement distress detection can be improved. Therefore, in the following, we first report the state-of-the-art approaches for distress detection. Afterward, we present related work for incrementally boosting neural networks and estimating a network's uncertainty in order to select data worth to be annotated for training.

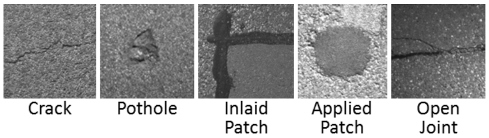

Fig. 3. Types of distress that can be detected.

A. Pavement Distress Detection

Automating distress detection has already attracted a lot of interest in the literature. Besides depth-based detection methods [3], [4] which are out of the scope of this paper, the main research is focused on image-based distress detection and can roughly be divided into two groups.

The first group of algorithms is focused on detecting cracks only and relies on the explicit assumption that crack structures can be identified as local intensity minima. Therefore, these methods are based on image thresholding, followed by crack refinement. The refinement algorithms are diverse. Some are based on morphological image operations and the search for connected components [5], [6], [7], [8]. Other approaches use graph-based crack candidate analysis for further refinement [9], [10], [11], [12], a multi-scale curvelet transform instead of a binary threshold [13], or Gabor filters in order to find crack candidates [14].

The algorithms of the second group apply different types of classifiers to patches of the image in order to extract crack or distress regions. For example, a support vector machine is applied to histograms of oriented gradients (HOG) features [15] or local binary patterns (LBP) [16], [17]. Convolutional neural networks have gained a lot of interest more recently. Starting with [18], a lot of different approaches with varying network structure have been proposed [1], [19], [20], [21].

Besides the different detection methods, the work presented in [22] already addresses the problem of presenting only these image patches to a human operator that are most crucial to increase the classifier quality. However, they have chosen the non-calibrated softmax output as a measure of the network's uncertainty which is inappropriate in most cases as explained in [23], [24] and further discussed in Sec. II-C.

Although a lot of different methods have been presented so far, there is a lack of publicly available datasets, with the majority of datasets only containing up to 500 images [8], [6], [12]. The German Asphalt Pavement Distress dataset (GAPs dataset) [1], that we made publicly available[2], is the first freely available pavement distress dataset of a size large enough to train high-performing deep neural networks. Although the dataset already has a decent size, it comprises only three different roads. Therefore, it still lacks the required diversity to represent all road surfaces and damage characteristics that are present in the German road network.

B. Incremental Boosting

One way to address a lack of data is incremental learning [25], [26], [27]. Incremental learning refers to continuously extending the network's knowledge by adapting to new data. In online incremental learning scenarios, new data are available only for a limited period of time due to memory restrictions. In contrast, we can access a persistent data storage containing lots of high-resolution road images from previous inspections. Unfortunately, these images are

[2] The GAPs dataset is publicly available at http://www.tu-ilmenau.de/neurob/data-sets-code/gaps/

provided with a coarse and partially inaccurate labeling, which is not sufficient to incrementally improve a neural network (for segmentation) in a supervised manner (see Sec. I). Therefore, a further challenge arises in our setting: We have to decide automatically which high-resolution road images are worth annotating by a human in the loop.

The overall process is closely related to active learning [28]. The primary goal of active learning is to train a neural network with as less labeled data as possible [29], [30]. The way to achieve this goal can be adapted to our scenario as well. In active learning, a system is created that is able to decide on its own which data to annotate and to process next. Following an initial training with a small amount of data, an acquisition function is used to select new samples based on the network's decisions. Usually, this acquisition function utilizes the network's uncertainty.

C. Uncertainty Estimation

By default, most deep learning based classifiers, as well as our ASINVOS net, are not capable to represent uncertainty. This is due to the parameter optimization with standard backpropagation, that results in a single best parameter configuration, containing only a point estimate for each network weight, and thus, throwing away any uncertainty information. Hence, the output of the final softmax layer, which is often treated as a class probability distribution, is rather a poorly calibrated mapping to a vector of relative class affiliations than a theoretically grounded measure of the network's certainty [23], [24]. Furthermore, recent work on adversarial examples [31], [32] has shown that neural networks are easily fooled. Applying imperceptible perturbations may lead to misclassification with erroneously high softmax output.

To obtain meaningful uncertainty estimates, it is necessary to consider the full posterior distribution over the network's weights. Unfortunately, exact Bayesian inference is intractable for complex neural networks. Therefore, several approaches have been proposed, approximating the posterior distribution by means of Markov Chain Monte Carlo [33], [34] or variational inference [35], [36], [37]. All these approaches have in common that they introduce additional overhead during training or do not scale properly in deep learning scenarios. With Dropout variational inference (DVI) [38], Gal and Ghahramani recently proposed the first practical tool to obtain uncertainty estimates even for complex networks. The only requirement for using DVI is the necessity of training the neural network with Dropout applied before each layer with weights. Dropout [39] is a frequently used regularization technique to reduce overfitting. When using Dropout, at each training step randomly selected neurons are removed from the network. This way, Dropout samples from an exponential number of different networks and prevents co-adaption. At test time, in contrast, all neurons are kept and the outgoing weights are scaled with respect to the Dropout probability in order to approximate an averaging effect similar to an ensemble. In [38], Gal and Ghahramani examine a further, so far unknown, link between Dropout and approximated Bayesian inference. They propose an interpretation of Dropout as Bernoulli approximated variational inference. This means, we can sample from the posterior distribution by applying Dropout at test time as well in order to obtain uncertainty information. For further details on DVI, we refer to [23], [40]. In addition, [41] proposes different measures to quantify the uncertainty, appropriate to be used as acquisition function.

Given an uncertainty estimate for each unknown, yet not accurately labeled, part of the image, we can easily decide whether it should be annotated by a human in the loop or not, reducing the labeling effort dramatically.

III. Incrementally Boosted Distress Detection

To incrementally boost the performance of our distress detector, we need to evaluate when the detector is uncertain in its decision. We decided in favor of Dropout variational inference since it allows us to extract theoretically grounded uncertainty estimates while being easy to integrate.

In this section, we first revisit our ASINVOS net, the convolutional neural network we conceptualized in [2] specifically for distress detection. Then, we describe the modifications needed to obtain uncertainty estimates, and discuss various measures to quantify the uncertainty.

A. Distress Detection

For distress detection, we conceptualized the ASINVOS net, a convolutional neural network with eight convolutional layers, three max-pooling layers, and three fully connected layers (see Fig. 4). The network architecture is inspired by the VGG-models [42] (spatial feature extraction using multiple units of two convolutional layers followed by one max-pooling layer) and AlexNet [43] (classification using fully connected layers with final softmax output). Except for the last layer, all neurons are ReLUs [44]. For implementation, we used Keras [45] based on Theano [46].

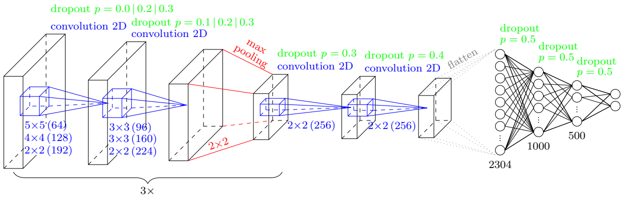

Fig. 4. Neural network for distress detection (ASINVOS net) [1]

The ASINVOS net has 4.0 M weights in total. Thus, regularization is the key to perform well on unknown data. Dropout is known to be a very good regularization technique that prevents co-adaption and also improves generalization abilities. Therefore, we make extensive use of Dropout before all weight layers except the input layer.

After patch-based training, the network is converted to a fully convolutional network as described in [47] in order to be applicable as segmentation network. Fig. 5 shows segmentation results of the ASINVOS net on unknown data. Obviously, it performs very well in segmenting road images, such that distress can be discovered easily. Nevertheless,

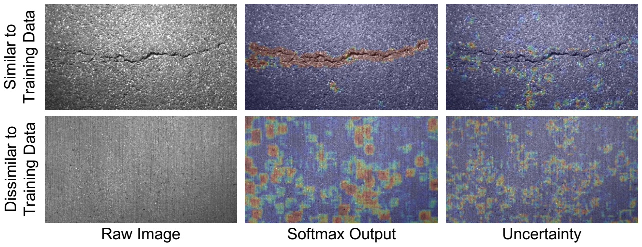

Fig. 5. Distress detection results on unknown data for images similar and dissimilar to training data. Besides the raw image and the network's output, the uncertainty in terms of variation ratio (see Sec. III-B) is depicted.

evaluations also showed, that the performance drops on road images, that are significantly different from training data.

B. Incremental Boosting

In order to select image subregions that will help to improve the performance of a neural network, we need to determine its uncertainty. A convenient technique to estimate a network's uncertainty is Dropout variational inference as explained in Sec. II-C.

According to [38], Dropout variational inference can be applied to every neural network that was trained with Dropout. Since our ASINVOS net already makes extensive use of Dropout before every weight layer, we do not need to modify its structure. Given a single patch $x \in \mathbb{X}$ of an input image, represented by a set of sliding-windows[3] $\mathbb{X}$, we can transform the uncertainty in the network's weights to predictive uncertainty by marginalizing over the approximate posterior distribution using Monte Carlo integration. [38] shows that this can be done by averaging multiple stochastic forward passes through the network with Dropout enabled at test time as well.

Given the output including predictive uncertainty, we need to quantify the uncertainty. In [41] various measures have been proposed, including intuitive measures, such as predictive entropy, mean standard deviation or variation ratio, and a more complex measure based on the mutual information between predictions and the model posterior distribution. In our setting, all measures performed similarly, with the variation ratio slightly ahead. Therefore, we decided in favor of the variation ratio as acquisition function. The variation ratio $r(x)$ describes a lack of confidence through the deviation from the mode c^* (most common predicted label):

$$r(x) = 1 - \frac{1}{N}\sum_{n=1}^{N}\delta(y^n) \text{ with: } \delta(y^n) = \begin{cases} 1, & y^n = c^* \\ 0, & y^n \neq c^* \end{cases} \tag{1}$$

with y^n being the predicted label at forward pass n.

Given an uncertainty estimate for each patch, and even more, for whole subregions of an input image, we can easily determine parts of an image that are worth annotating by a human in the loop by repetitively selecting those patches with the highest variation ratio. Since we do not have label information in advance, we select patches that maximize the acquisition function solely.

IV. Integration into Image Acquisition Process

In order to assess the usefulness of image subregions, the high-resolution road images have to be preprocessed first. Following German federal regulations, each road image contains ten meters of a road composed of several HD pictures. Unfortunately, during previous inspections, the lane's position in each image was determined manually by the operator in order to fit the label grid (Fig. 2), but not stored. To ensure that only valid image patches can be selected as training data, we have to detect the lane again. Patches outside the lane may contain non-road structures like grass or gravel. Since these structures are excluded from training, the neural network would be uncertain in its decisions and preferably select these subregions for annotation. Including these regions would complicate the decision process and may worsen the performance of the neural network. Furthermore, as shown in Fig. 2, the road images contain clearly visible stitching edges. Patches at stitching edges may be confused with cracks and should be excluded as well. Therefore, in the following, we describe how we preprocess the high-quality road images to ensure that only useful patches are selected during incremental boosting.

A. Detecting Stitching Edges

Following German federal regulations, the single images that compose a high-resolution road image have to be stitched unaltered. Therefore, the radially symmetrical light drop-off in the single images causes different light intensities at stitching edges. These artificial structures are likely to distract the classifier, and therefore, need to be detected and handled separately.

Since stitching edges occur almost regularly within a high-quality road image, the basic idea is to use frequency analysis to find them. We take this into account to keep our approach as simple as possible in order to ensure fast processing times. For detecting the vertical edges, we apply a vertically oriented Sobel filter to the entire image, and sum up the results within every single row to obtain a one-dimensional edge candidate vector. This vector is transformed into the frequency domain using a fast Fourier transform, and the maximum amplitude within a reasonable frequency spectrum is obtained. This maximum is transformed back to the pixel domain in order to set up a minimum distance between two stitching edges. The final edge detection is performed by repeatedly taking the maximum of the edge candidate vector and setting the surrounding pixels at a distance smaller than the minimum distance to zero. To further improve the results,

[3] Due to the conversion to a fully convolutional network, our ASINVOS net is able to process images of any size. Each pixel in the final output volume corresponds to a region of 64×64 pixels in the input image as this was the input shape during patch-based training. Since we do not use zero padding in convolutional layers, this fully convolutional approach is equivalent to applying a sliding-window of size 64×64 pixels with a stride equal to the product of the strides in all spatial layers. However, the segmentation approach is several times faster due to the elimination of redundant computations. For simplification, all explanations in this paper rely on the sliding-window approach, but the extension to entire images is straightforward.

the vertical edge detection is performed separately for the left- and right side of the image. If the extracted stitching positions disagree due to heavily structured elements like gullies, the result with a higher regularity is favored.

Horizontal edges can be detected in an analogous manner. Since the horizontal stitching position was fixed and known in advance for the data used in the experimental section, this step could be omitted.

B. Lane Detection

To detect the lane, we need to extract information about limiting structures. For example, road markings, curbs, sidewalks, bikeways and other limiting elements are of relevance. Since these structuring elements occur in various appearances, we decided in favor of a neural network to robustly detect them. Luckily, we had a neural network, already trained on these data, so we were able to apply transfer learning. We adapted the ASINVOS net architecture (see Sec. III-A, Fig. 4) by changing only the output coding. This convolutional neural network is able to distinguish between road markings, roadsides, and regular road (Fig. 6). It was trained on the GAPs dataset (see Sec. II-A) with additional annotations for road markings and roadsides. After training, it was converted to a fully convolutional network for image segmentation.

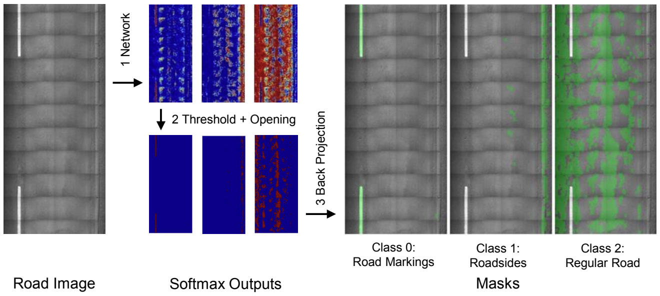

Fig. 6. Workflow to distinguish regular road from road markings and roadsides. First, an adapted ASINVOS net is applied. Subsequently, to get an accurate segmentation, the network's output is postprocessed and upsampled considering the network's properties.

To detect the limits of the lane, we follow the workflow shown in Fig. 6: First, we apply the neural network to the image. Then, a threshold is applied to get binary masks. Finally, these masks are upsampled considering both the input shape and the overall stride of the neural network. Since the mask representing regular road is not sufficient to detect the entire lane, we follow a fusion strategy. The masks representing road markings and roadsides are fused as shown in Fig. 7 to get a single mask of lane limiting structures. Road markings can appear at every part of the image, while roadsides are expected to be at the outer borders of the image. Both facts are taken into consideration in our fusion strategy.

Given the fused mask containing all lane limiting elements, we can estimate the lane's position. First, we sum up the elements along the driving direction to get the cross section of the road. Next, the cross section is divided into several segments using a threshold. Since road markings, like arrows, may lead to disconnected segments, we start from the largest central segment and follow a rule-based fusion strategy incorporating the specifications for legitimate roadway widths to finally detect the lane.

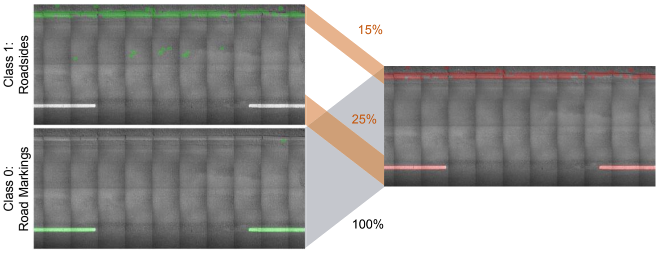

Fig. 7. Workflow for combing the masks representing road markings and roadsides. While roadsides are supposed to appear at the outer borders only, road markings can appear everywhere due to road narrows on both sides.

V. Experiments

In this section, we evaluate the performance of our proposed system to incrementally boost pavement distress detection with a human in the loop. We start by evaluating the preprocessing steps since they are both a necessary prerequisite to be able to use high-resolution road images from previous inspections. Then, we conduct experiments on our GAPs dataset to evaluate the idea of incremental learning using patches the network is most uncertain about. Finally, we show how the overall system can be used to incrementally boost the performance of our ASINVOS net using road images from previous inspections.

A. Preprocessing

For evaluating the preprocessing steps, we used 1,637 manually annotated high-resolution road images, showing ten meters of a lane each. The images are taken from several German federal highways. Since preprocessing steps are not the main focus of this paper, we only briefly report the respective results in the following.

1) Detecting Stitching Edges: Fig. 8 shows visualized results for stitching edges detection. In 1,610 images, all edges could be detected correctly (98.35%). An error analysis showed that in six images, a single edge at the very left or right was missing. 20 images showed errors in the middle due to image structures from gully covers and the like. In one image the detection failed completely.

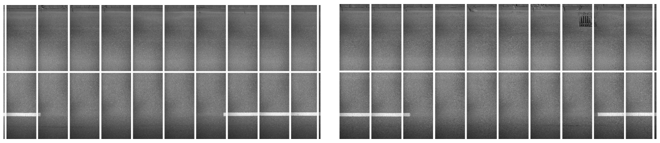

Fig. 8. Results for stitching edges detection. Heavily structured elements like gullies (right) do not cause problems.

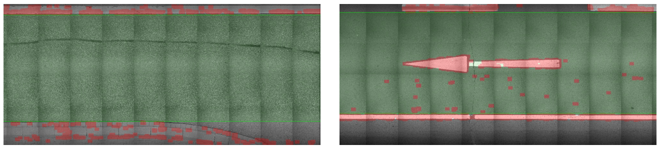

Fig. 9. Results for lane grid detection (green). Road markings in the middle of the lane (right) can be handled correctly. The mask showing potential lane limits is highlighted red.

2) Lane Detection: Fig. 9 shows exemplary results for lane detection. In 1,541 images, the lane could be detected correctly (94.14%). Road markings in the middle of the lane can be handled correctly. Errors were mainly caused by faded road markings leading to missed lane limits.

Additionally to the detection results, we recorded certainty measures to judge if an image is suitable for further processing. According to this evaluation, more than 90% of recorded high-resolution road images can be used for incremental boosting.

B. *Incremental Boosting*

For evaluating the proposed system, we conducted several experiments on our GAPs dataset. In order to make the experiments feasible at all, we used a subset of the training and validation set only. For training and validation, we randomly sampled 50,000 and 10,000 patches, respectively. To further enhance the expressiveness of the subsets, sampling was done at a ratio of six patches showing intact road to four patches showing distress.

In all experiments, we used stochastic gradient descent with a fixed learning rate of 0.01, momentum of 0.9, and weight decay of 0.00005. At each training step, we presented a batch containing 256 patches while randomly flipping them horizontally and vertically. In order to obtain meaningful uncertainty estimates, we follow the recommendations in [41] and used 100 stochastic forward passes for each patch. Nevertheless, 25 stochastic forward passes already seem to perform similarly.

To ensure reporting meaningful results only, we repeated each experiment three times with different random seeds for Dropout and sample presentation order, averaging the results.

1) Evaluating the Idea of Incremental Learning: In a first experiment, we evaluated whether it is possible to improve the performance of our ASINVOS net incrementally. To do so, we first trained five reference classifiers with 10,240, 20,480, 30,720, 40,960, and 50,000 randomly selected patches[3], respectively. For all reference classifiers, the best weights configuration was chosen within 400 epochs based on the performance on the validation set. Next, we trained classifiers while incrementally expanding the training set from 10,240 patches to 50,000 patches. To ensure a stable weight initialization, we always used the weights of the reference classifier trained with 10,240 patches as initial weights configuration. In each step, we added 1,024 patches[4] to the training set and trained the classifier for further 40 epochs. For selecting new patches, we compared our acquisition function to random choices.

Furthermore, to examine a potential loss of precision due to incremental training, we repeated the experiment and trained the classifiers after each acquisition step for 400 epochs from scratch. Results of this first experiment are depicted in Fig. 10.

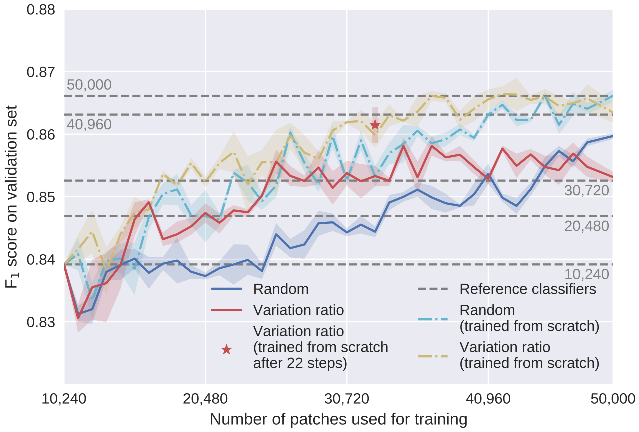

Fig. 10. Performance on the validation set while incrementally expanding the training set with patches taken from remaining GAPs training subset. Patches are selected either randomly or based on the variation ratio. Classifiers trained from scratch after each acquisition step are depicted with dash-dotted lines. For further details, we refer to Sec. V-B.1.

Fig. 10 shows that the performance of our ASINVOS net can be improved incrementally. Furthermore, it is obvious that selecting new patches based on the variation ratio constantly leads to better results. Both incrementally trained classifiers (solid lines in Fig. 10) do not reach the performance of the reference classifier trained with all 50,000 samples. The results for the classifiers always trained from scratch (dash-dotted lines in Fig. 10), show that this is most likely due to the incremental learning. Unfortunately, training from scratch is not feasible when dealing with large datasets. However, we later show how to choose suitable points to trigger a full retraining.

Independently of the way the classifier is trained, we observed that, when selecting new patches based on the variation ratio, the performance remains the same and does not increase anymore after about 22 acquisition steps (32,768 patches in total). Fig. 11 visualizes the mean variation rate at each acquisition step. All uncertainty seems to be explained away after 22 acquisition steps since the mean variation rate reaches almost zero. In subsequent steps, only redundant samples not providing any additional information, are added. Thus, the performance remains the same.

Consequently, we conclude that we could have selected a better subset with less patches based on the variation rate. Furthermore, as both curves have a similar trend, this is a good point to trigger a full retraining. As shown in Fig. 10, the retrained classifier (red star in Fig. 10) reaches the performance of the classifier that was always retrained from scratch.

[3] multiples of the batch size (256), except for the last one

[4] 1,024 patches are equivalent to 4 new batches of 256 patches

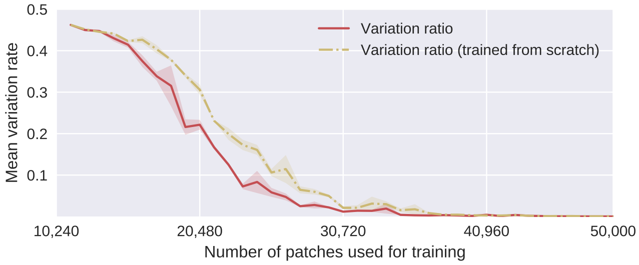

Fig. 11. Mean variation ration at each acquisition step as a function of the number of patches used for training both the incremental classifier and the classifier always trained from scratch.

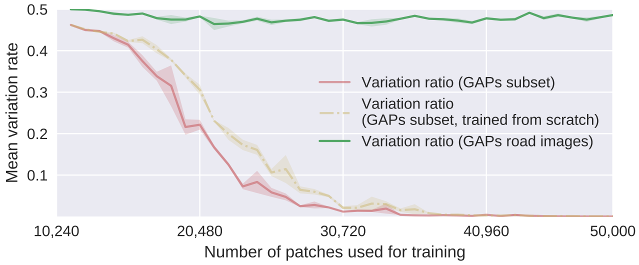

Fig. 13. Mean variation ratio at each acquisition step as a function of the number of patches used for training for all incrementally trained classifiers in comparison.

2) Application to high-resolution Road Images: Now that we know we are able to obtain meaningful uncertainty estimates, we can apply the overall system to high-resolution road images from previous inspections. Unfortunately, in such an early stage of development, experiments with real humans are not feasible. Therefore, we made use of the high-resolution road images that have been used to create the GAPs dataset. These images have been annotated manually by trained operators such that an actual damage is enclosed accurately by a bounding box, and thus, are perfectly suitable for simulating a human in the loop.

Again, we trained classifiers while incrementally expanding the training set from 10,240 patches to 50,000 patches. In contrast to the previous experiment, new patches are directly sampled from entire images. Thus, we have to assign the label automatically. A patch is labeled as distress candidate, if at least a region of 32×32 pixels in the center of the patch was annotated as distress. In each step, we applied the network to three randomly selected high-resolution road images in order to select new patches and trained the classifier for further 40 epochs. We decided in favor of selecting a fixed number of 1,024 patches again, to be able to compare the results to the previous experiment. However, selecting patches up to a minimum threshold for the variation rate is also possible. Results of this experiment are depicted in Fig. 12.

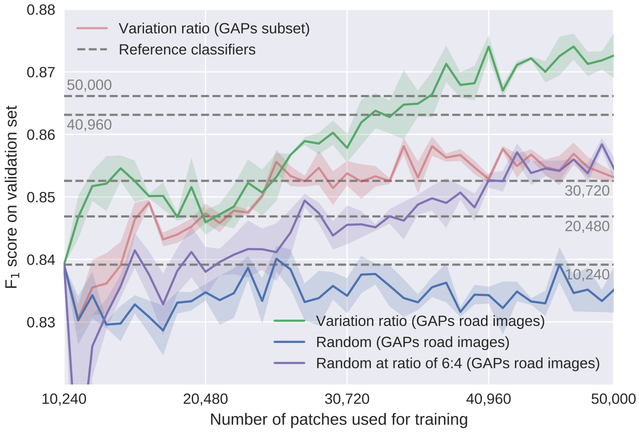

Fig. 12. Performance on validation set while incrementally expanding the training set with patches taken from entire high-resolution road images of the GAPs dataset. Patches are selected either randomly or based on the variation ratio. For comparison, the best classifier incrementally trained on GAPs training subset is depicted as well.

Fig. 12 shows that the performance of our ASINVOS net can be improved constantly when selecting patches based on the variation ratio. Randomly selecting new patches performed worse even if we forced the same ratio of six patches showing intact road to four patches showing distress as in the initial training set. Furthermore, since we are not limited to the randomly drawn subset of 50,000 patches, the improvement continues after 22 acquisition steps as well. As shown in Fig. 13, this constant increase is caused by the ongoing selection of patches the network is most uncertain about.

VI. Conclusion

We showed how the performance of a neural network for pavement distress detection can be improved with a human in the loop. The basic idea is to use a large number of images from previous inspections for retraining. However, this includes an annotation step, that is labor-intensive. Therefore, we showed how to select data, that are worth annotating. In order to decide which images would help to improve the performance, we trained a neural network such that we can determine what it does not know and utilized its uncertainty information for selecting new training patches. Using this technique, we were able to significantly improve the performance of our ASINVOS net with only a fraction of data that would usually be necessary to get the same performance.

Using this approach, as part of the ASINVOS project, our industrial partner enlarged the GAPs dataset with images from many types of German federal roads. This enables the employment of the neural network on a larger variety of roads, and thus, helps to further automate the yet labor-intensive road inspection process.

In future work, we plan to integrate more advanced techniques for uncertainty estimation, like Concrete Dropout [48]. This may improve the selection of training images, and thus, may allow for annotating even less data.

References

[1] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.-M. Gross, "How to get pavement distress detection ready for deep learning? a systematic approach." in *Int. Joint Conf. on Neural Networks (IJCNN)*, 2017, pp. 2039–2047.

[2] M. Eisenbach, R. Stricker, K. Debes, and H.-M. Gross, "Crack detection with an interactive and adaptive video inspection system," in *Arbeitsgruppentagung Infrastrukturmanagement*, 2017, pp. 94–103.

[3] T. Yamada, T. Ito, and A. Ohya, "Detection of road surface damage using mobile robot equipped with 2D laser scanner," in *Int. Symp. on System Integration (SII)*. IEEE/SICE, 2013, pp. 250–256.

[4] Y. Yu, H. Guan, and Z. Ji, "Automated detection of urban road manhole covers using mobile laser scanning data," *Trans. on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3258–3269, 2015.

[5] L. Peng, W. Chao, L. Shuangmiao, and F. Baocai, "Research on crack detection method of airport runway based on twice-threshold segmentation," in *Int. Conf. on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*. IEEE, 2015, pp. 1716–1720.

[6] H. Oliveira and P. L. Correia, "CrackITan image processing toolbox for crack detection and characterization," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2014, pp. 798–802.

[7] K. Xu, N. Wei, and R. Ma, "Pavement crack image detection algorithm under nonuniform illuminance," in *Int. Conf. on Information Science and Technology (ICIST)*. IEEE, 2013, pp. 1281–1284.

[8] S. Chambon and J.-M. Moliard, "Automatic road pavement assessment with image processing: review and comparison," *Int. Journal of Geophysics*, vol. 2011, 2011.

[9] H. Oliveira and P. L. Correia, "Road surface crack detection: Improved segmentation with pixel-based refinement," in *Europ. Signal Processing Conf. (EUSIPCO)*. IEEE, 2017, pp. 2026–2030.

[10] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, "Automatic crack detection on two-dimensional pavement images: an algorithm based on minimal path selection," *Trans. on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2718–2729, 2016.

[11] J. Tang and Y. Gu, "Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis," in *Int. Conf. on Systems, Man, and Cybernetics (SMC)*. IEEE, 2013, pp. 3026–3030.

[12] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.

[13] G. Wu, X. Sun, L. Zhou, H. Zhang, and J. Pu, "Research on crack detection algorithm of asphalt pavement," in *Int. Conf. on Information and Automation*. IEEE, 2015, pp. 647–652.

[14] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the gabor filter," in *Int. Conf. on Intelligent Transportation Systems (ITSC)*. IEEE, 2013, pp. 2039–2044.

[15] R. Kapela, P. Śniatała, A. Turkot, A. Rybarczyk, A. Pożarycki, P. Rydzewski, M. Wyczałek, and A. Błoch, "Asphalt surfaced pavement cracks detection based on histograms of oriented gradients," in *Int. Conf. on Mixed Design of Integrated Circuits & Systems (MIXDES)*. IEEE, 2015, pp. 579–584.

[16] M. Quintana, J. Torres, and J. M. Menéndez, "A simplified computer vision system for road surface inspection and maintenance," *Trans. on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 608–619, 2016.

[17] S. Varadharajan, S. Jose, K. Sharma, L. Wander, and C. Mertz, "Vision for road inspection," in *Winter Conf. on Applications of Computer Vision (WACV)*. IEEE, 2014, pp. 115–122.

[18] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2016, pp. 3708–3712.

[19] Z. Fan, Y. Wu, J. Lu, and W. Li, "Automatic pavement crack detection based on structured prediction with the convolutional neural network," *arXiv:1802.02208*, 2018.

[20] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes, "Deeper networks for pavement crack detection," in *Int. Symp. on Automat. and Robotics in Construction (ISARC)*, vol. 34. IAARC, 2017, pp. 479–485.

[21] X. Wang and Z. Hu, "Grid-based pavement crack analysis using deep learning," in *Int. Conf. on Transportation Information and Safety (ICTIS)*. IEEE, 2017, pp. 917–924.

[22] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials*, vol. 157, pp. 322–330, 2017.

[23] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.

[24] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Int. Conf. on Machine Learning (ICML)*, 2017.

[25] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," in *Europ. Symp. on Artificial Neural Networks (ESANN)*, 2016.

[26] C. Karaoguz and A. Gepperth, "Incremental learning for bootstrapping object classifier models," in *Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1242–1248.

[27] C. Kding, E. Rodner, A. Freytag, and J. Denzler, "Fine-tuning deep neural networks in continuous learning scenarios," in *ACCV Workshop on Interpretation and Visualization of Deep Neural Nets (ACCV-WS)*, 2016.

[28] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, no. 55-66, p. 11, 2010.

[29] X. Li and Y. Guo, "Adaptive active learning for image classification," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 859–866.

[30] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," *NIPS Workshop on Bayesian Deep Learning*, 2016.

[31] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[32] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. corr abs/1412.1897 (2014)," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Int. Conf. on Machine Learning (ICML)*, 2011, pp. 681–688.

[34] C. Li, C. Chen, D. Carlson, and L. Carin, "Preconditioned stochastic gradient langevin dynamics for deep neural networks," in *Conf. on Artificial Intelligence (AAAI)*, 2016.

[35] A. Graves, "Practical variational inference for neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 2348–2356.

[36] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Int. Conf. on Machine Learning (ICML)*, 2015, pp. 1613–1622.

[37] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 2575–2583.

[38] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. on Machine Learning (ICML)*, 2016, pp. 1050–1059.

[39] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[41] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," *Workshop on Bayesian Deep Learning, NIPS 2016*, 2016.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. on Learning Representations (ICLR)*, 2015, pp. 1–14.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[44] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Int. Conf. on Machine Learning (ICML)*, 2010, pp. 807–814.

[45] F. Chollet, "Keras," https://github.com/fchollet/keras, 2018.

[46] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv:1605.02688*, 2016.

[47] M. Eisenbach, D. Seichter, T. Wengefeld, and H.-M. Gross, "Cooperative multi-scale convolutional neural networks for person detection," in *Int. Joint Conf. on Neural Networks (IJCNN)*. IEEE, 2016, pp. 267–276.

[48] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *Advances in Neural Information Processing Systems*, 2017, pp. 3584–3593.