Improving Visual Road Condition Assessment by Extensive Experiments on the Extended GAPs Dataset

Ronny Stricker[a], Markus Eisenbach[a], Maximilian Sesselmann[b], Klaus Debes[a], and Horst-Michael Gross[a]

[a] Ilmenau University of Technology
Neuroinformatics and Cognitive Robotics Lab
98684 Ilmenau, Germany

[b] LEHMANN + PARTNER GmbH
99086 Erfurt, Germany

ronny.stricker@tu-ilmenau.de

***Abstract*—Aging public roads need frequent inspections in order to guarantee their permanent availability. In many countries, this includes the standardized visual assessment of millions of images. Due to the lack of sophisticated approaches, often, the evaluation is done manually and therefore requires excessive manual labor. GAPs is the most extensive publicly available dataset that provides standardized, high-quality images for training deep neural networks for pavement distress detection. We further enlarge this dataset and provide refined annotations. By conducting extensive experiments on the GAPs dataset, we improve the performance of automated visual road condition assessment. We evaluate the performance gain of several modern neural network architectures and advanced training techniques.**

***Index Terms*—Pavement distress detection, Convolutional Neural Networks, Deep learning dataset**

I. Introduction

Public infrastructures are suffering from aging and therefore need frequent inspection. Distress detection and solid management for maintenance are the keys to guarantee their permanent availability. Therefore, condition acquisition and assessment must be applied to the whole road network of a country frequently.

Following German federal regulations, the surface characteristics have to be evaluated i.a. regarding substance condition. The substance condition is captured with camera systems and has to be evaluated by visual inspection of the recorded images. Current evaluation is done manually and therefore requires excessive manual labor. This includes tasks like finding very thin cracks, that appear only in few pixels of the image. Therefore, the period between the actual inspection and the final evaluation may be up to several months. In the meantime, small damages, like cracks, can lead to substantial downtimes with a high impact on the population.

In the research project ASINVOS[1], we aim to automate this process to a high degree by applying machine learning techniques. The basic idea is to train a self-learning system with manually annotated data from previous inspections so that the system learns to recognize underlying patterns of distress. Once the system can identify intact infrastructure robustly, it can reduce the human amount of work by presenting only distress candidates to the operator. This helps to speed up the inspection process significantly and simultaneously reduces costs. Furthermore, inspection intervals can be reduced, which helps to remedy deficiencies in time.

Therefore, in [1] we presented the German Asphalt Pavement Distress (GAPs) dataset, which was the first high standard dataset in the pavement distress domain that provides high-quality standardized images and is large enough to allow the training of deep neural networks with high accuracy. This dataset already attracted much attention and had been used by several research groups from different countries, e.g. [2] and [3]. Due to many requests, as part of this paper, we enlarge the dataset by 500 additional images from an additional Federal highway. We also refined the annotations in order to provide data with even higher quality. Additionally, we created an MNIST- or CIFAR-like subset consisting of 50k samples such that a faster training is possible. This enables research groups to compare with the state of the art at a low cost.

In this paper, we conducted extensive experiments on the GAPs dataset. This includes the evaluation of several state-of-the-art network architectures and training techniques known to work well on other computer vision datasets. Additionally, we analyzed questions like "How many context is necessary for distress detection?" and "Which network complexity is necessary to perform at a high level and reasonable speed?".

*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: Assistierendes und Interaktiv lernfähiges Videoinspektiossystem für Oberflächenstrukturen am Beispiel von Straßenbelägen und Rohrleitungen (Interactive machine learning based monitoring system for pavement surface analysis)

II. Related Work

With first attempts published in the early nineties e.g., [4], automating the distress detection process has already been addressed by researchers for almost three decades. Therefore, a wide variety of different approaches have been presented ranging from traditional image processing techniques to deep learning approaches more recently. Since the typical pavement assessment processes are carried out using 2D image recordings, the related work section is focused on 2D image processing approaches. However, some authors have shown the effectiveness of using 3D data for distress detection [5]. We refer to [6] for a more detailed list of depth based 3D approaches.

The algorithms developed for 2D image-based evaluation of the pavement surface can be divided into two major categories:

- computer vision algorithms designed explicitly for crack detection, mostly by applying image value thresholding and
- algorithms for general distress detection that use implicit or explicit local feature extraction.

A. Crack Detection

The first group of algorithms uses image processing methods to detect road distress structures that can be extracted by thresholding afterward. Therefore, preprocessing algorithms are applied in order to reduce illumination artifacts. Under the assumption that crack structures can be identified as local intensity minima, this group of algorithms usually uses thresholding in the image space to find crack candidates. The resulting crack image is further refined by morphological image operations and by searching for connected components. Approaches belonging to this category are presented in [7], [8], as well as in [9], where the closed source but publicly available *CrackIT* toolbox is presented. Other variants of this category, e.g. use minimal-path-based [10] or graph-based [11] crack candidate analysis for further refinement, that is also used by the well known CrackTree approach [12]. Although most authors focus on crack detection, thresholding techniques are also used for pothole detection [13].

B. Feature-based Distress Classification

The algorithms of the second category apply classifiers to local regions of the image in order to extract crack or distress regions. Traditional image processing approaches mostly apply explicit feature extraction. Using a Support vector machine (SVM) is very common among these traditional approaches. For example, this classifier is applied to Histogram of Oriented Gradient (HOG) features [14] or Local Binary Patterns (LBP) [15], [16]. A recently proposed approach has been applied to frontal-facing images by integrating an image clustering step to extract the street surface first of all [17].

More recent approaches tend to use implicit feature-extraction by using *Convolutional Neural Networks (CNNs)*. Approaches based on CNNs mainly differ regarding network architecture, predicted distress classes and whether downward or frontal-facing input images are processed.

One of the first attempts to apply CNNs in the domain of pavement distress detection is presented in [18]. The network architecture shares some similarities with LeNet-5 [19] but utilizes ReLU activation and four convolution/pooling layers for detection pavement cracks on downward facing road images. Also applied to downward facing road images, but utilizing VGG-based CNNs [20] are the approaches presented in [21] and [1]. While the former approach is focused on crack detection, the latter addresses all distress types addressed by the federal German pavement assessment process.

With the tremendous distress detection improvements over traditional image processing techniques archived by CNNs [1], [12], distress detection in frontal-facing images it getting more common. This kind of image is often processed by a further processing step to constrain distress detection to the pavement area. This step can be carried out using traditional image segmentation techniques like graph-based hierarchical clustering [22] or using CNNs like SegNet [3]. The network architectures used for processing frontal-facing images are based on state-of-the-art image processing networks. [23] for instance compares InceptionV2 and MobileNet for the detection of eight different distress classes while [3] applies Squeeze-Net for distress detection. [2] present a Feature Pyramid and Hierarchical Boosting Network which is used for crack detection in frontal-facing images.

In this paper, we focus on distress detection in downward facing road images. Frontal-facing images do not provide the resolution necessary to detect minimal damages. This can be seen in many approaches trained on frontal-facing images, since they often miss tiny cracks, that are still detected by approaches applied to downward facing road images.

C. Datasets

Although a lot of different methods have been presented so far, there is still a lack of publicly available datasets that are of decent size and are recorded in a standardized way. This hampers comparability since most publications are using own datasets that have been generated using consumer cameras and are labeled in different ways.

The datasets published so far do often consist of less than 500 images, e.g., [2], [8], [9], [12], and do not offer the necessary diversity to train a universal pavement distress detector. Although, some datasets have been released recently that do offer a decent size like [24] with 700k Google Street View images or [23] with 15k smartphone images, these datasets are using frontal-facing images and do not provide the level of resolution required for formal road assessment.

III. Dataset

The GAPs dataset[2] is the most extensive dataset in the pavement distress domain that provides standardized, high-quality images. Due to many requests, we provide 500 additional images from an additional Federal highway to enlarge the dataset even more. Now, it consists of 2468 HD road surface

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

images. We also provide a sub-sampled dataset containing 50k images to allow for fast training and comparison to the state of the art in an MNIST- or CIFAR-like fashion. Most important, we improved the annotations. The level of detail is increased by labeling distress by several small bounding boxes that enclose the distress tightly. Additionally, all annotations were checked for correctness by several experts. In the following, we refer to this dataset as GAPs v2.

A. Standardized Data Acquisition

The data acquisition is based on the specification by the German Road and Transportation Research Association (FGSV) – the so-called Road Monitoring and Assessment (RMA) [25]. The image data of the GAPs dataset have been captured by the mobile mapping system S.T.I.E.R (Fig. 1), that is certified annually by the German Federal Highway Research Institute (BASt) since 2012. This vehicle is equipped with several high-resolution cameras, i.a. two slightly overlapping bird-eye-view photogrammetrically calibrated monochrome cameras capturing the pavement's surface in detail. The surface camera system is synchronized with a high-performance lighting unit. This allows continuous capturing of road surface images even at high velocities (ca 80 km/h) and independent of the natural lighting situation. The cameras capture images at a resolution of 1920×1080 pixels, which means each pixel covers $1.2\,\text{mm}\times1.2\,\text{mm}$ of the surface. For more details regarding the data acquisition process and the measurement vehicle, we refer to [1]. Following the German FGSV-regulations, the surface defect classes shown in Fig. 2 must be detected.

B. Improvement of the GAPs dataset

The GAPs v2 dataset[3] is a significant improvement over the original GAPs dataset [1]:

[3]The GAPs v2 dataset is available at http://www.tu-ilmenau.de/neurob/data-sets-code/gaps/.

Fig. 1: Mobile mapping system S.T.I.E.R

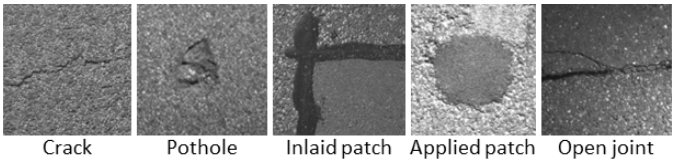

Fig. 2: Surface defect classes in GAPs dataset. The class cracks comprises all sorts of cracks like single/multiple cracking, longitudinal/transversal cracking, alligator cracking, and sealed/filled cracks.

TABLE I: Class distribution of GAPs dataset.

(a) Full dataset

Class	Fraction
Intact road	89.71%
Cracks	7.28%
Applied patches	1.72%
Inlaid patches	0.75%
Potholes	0.30%
Open joints	0.24%

(b) 50k dataset

Class	Fraction
Intact road	60%
Cracks	20%
Applied patches	10%
Inlaid patches	5%
Potholes	3%
Open joints	2%

1) More data: The GAPs v2 dataset includes a total of 2 468 gray valued images (8 bit), partitioned into 1 417 training images, 51 validation images, 500 validation-test images, and 500 test images, following the partitioning suggestions of [26]. Using these images, 692 377 patches of surface defects and 6 035 404 patches of intact road are extracted. Tab. I(a) shows the unbalanced class distribution of the full dataset. The pictured surface material now contains pavement of four different German federal roads.

2) Refined annotations: The images have been annotated manually by multiple trained operators at a high-resolution scale such that actual damage is enclosed by a bounding box and the non-damage space within a bounding box has a size of lower than 64×64 pixels. All annotations of the first version of the GAPs dataset have been refined, such that the non-damage space within a bounding box is even smaller than that restriction. Conflicting annotations have been resolved (GAPs v1 had only one annotator per image).

3) More context: While GAPs v1 offered only patches of size 64×64 extracted within the annotated regions and the intact surface regions, GAPs v2 offers several patch sizes showing more context (see Fig. 4). The defect region is still ensured to be within the 64×64 center region of each patch, but the surroundings may help to make correct decisions. In Sec. V we show the benefits of this context information.

4) 50k subset available: Since deep learning benefited most from small size real-world datasets, we also created a smaller subset for fast experiments. Inspired by the MNIST and CIFAR datasets, we created a training set of 50 000 samples. Additionally, the validation set, validation-test set, and test set contain 10 000 samples each. Tab. I(b) shows the chosen class distribution of the 50k subset. The samples for each class were chosen randomly until the desired number of samples was reached. The classes are left unbalanced, but the dominant classes of intact road and cracks are not that dominant as in the full dataset. The relative fraction of the non-dominant classes

TABLE II: ASINVOS net for patches of size 160×160.
Abbreviations: D – dropout (dropout probability), in – input, conv – convolution, pool – max pooling, fc – fully connected layer, out – softmax output

type	filter size	stride	regular.	output size	# paramet.
in				1×160×160	
conv	5×5 (64)	1×1	—	64×156×156	1 664
conv	3×3 (96)	1×1	D (0.1)	96×154×154	55 392
pool	2×2	2×2		96×77×77	
conv	4×4 (128)	1×1	D (0.2)	128×74×74	196 736
conv	3×3 (160)	1×1	D (0.2)	160×72×72	184 480
pool	2×2	2×2		160×36×36	
conv	3×3 (192)	1×1	D (0.3)	192×34×34	276 672
conv	3×3 (224)	1×1	D (0.3)	224×32×32	387 296
pool	2×2	2×2		224×16×16	
conv	3×3 (256)	1×1	D (0.3)	256×14×14	516 352
conv	3×3 (256)	1×1	D (0.4)	256×12×12	590 080
pool	2×2	2×2		256×6×6	
conv	3×3 (256)	1×1	D (0.4)	256×4×4	590 080
conv	2×2 (256)	1×1	D (0.4)	256×3×3	262 400
flat				2304	
fc	(1000)		D (0.5)	1000	2 305 000
fc	(500)		D (0.5)	500	500 500
out	(2)		D (0.5)	2	1 002
					5 867 654

among themselves are similar to the original dataset. We have chosen this distribution in order to focus more on the distress than on the intact road. The experiments in Sec. V confirm, that this approach is an excellent choice since observations for classifiers trained on the small subset transfer to identical classifiers trained on the full dataset.

IV. Deployed Network Architectures

A. AsinvosNet

As a baseline, we use the ASINVOS net that was the best performing neural network on GAPs in [1]. The ASINVOS net was designed for processing patches of size 64×64. It has eight convolutional layers, three max-pooling layers, and three fully connected layers resulting in 4.0 M weights and is similar to a VGG-model [27]. In order to process larger patch sizes, we slightly modified the convolutional part of the network, such that the first fully connected layer is applied to the identical number of 2304 neurons as in the 64×64 patch size version. Tab. II shows the filter sizes of the ASINVOS net for processing inputs of size 160×160. It has ten convolutional layers, four max-pooling layers, and identical fully connected layers as the original ASINVOS net. In sum, the modified ASINVOS net has 5.9 M weights. Thus, dropout is needed as regularization to perform well on unknown data.

B. Residual Networks

The ASINVOS net needs extensive regularization to compensate for a large number of weights in order to avoid overfitting of the training data. Furthermore, the high number of weights in combination with a high dropout rate requires many training epochs to adapt all network weights.

To overcome these problems, we opted for Residual Networks (ResNets) as proposed in [28], which are often referred to as *pre-activation Residual Networks*. This is a modern network architecture that speeds up training and inference

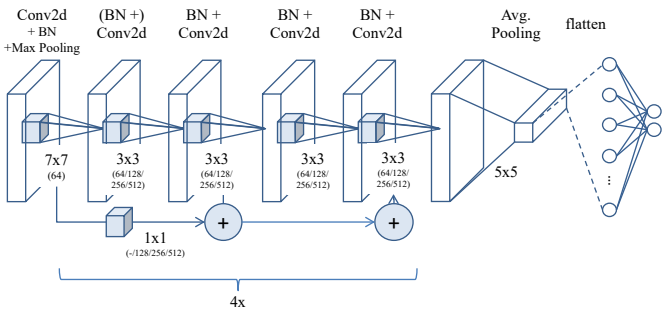

Fig. 3: Structure of the used 18 layer Residual Network. The shortcut connections bypass the residual blocks and improve gradient flow. The network consists of 18 convolutional layers, while every ResNet-block consits of two layers.

compared to the VGG-based networks used in [1]. The basic idea of ResNets is to introduce residual connections. This connections act as a bypass for blocks of convolutional layers (ResNet-blocks) and connect the output of the previous block with the output of the current block. Since the output of the previous block and the current block are added (See Fig. 3), the current block can concentrate on solving the residual and is therefore often called *Residual Block*. Furthermore, the introduced shortcut connections improve the gradient flow through the network an enables training of deeper networks.

C. Transfer Learning

Convolutional Neural Networks are known for learning features that generalize well between different tasks. Therefore, weights of a CNN trained on one task can be used for initializing a CNN to be trained on another task. This is known as transfer learning. In this paper, we use weights of a ResNet50 trained on ImageNet.

However, ImageNet provides colored (3-channel) images as input for the CNN whereas GAPs provides gray-value (1-channel) images. Thus, the weights of the first convolutional filter need to be adapted. We sum the weights for each position and output dimension channel-wise in order to process 1-channel images. The result is equal to triplicate the gray-value channel as 3-channel input. Since gray-value (3-channel) images are included at a small portion in ImageNet, at least the textural filters in the first layer should be reusable.

Another difference is the number of output classes in ImageNet and GAPs. Since we cannot map any of the ImageNet classes on the distress types, the weights of the last layer are likely useless. Thus, we initialize these weights randomly.

D. Adversarial Training

This technique initially was proposed to withstand adversarial attacks that try to fool a neural network. The basic idea is to manipulate samples of the dataset such that the CNN would make false decisions and use these adversarial samples additionally for training. This can be done directly by modifying the loss with an additional backward and forward pass without the need for actually creating the modified

samples. We follow the approach of [29] by using the fast gradient sign method (FGSM).

In our case, the primary goal is not to withstand adversarial attacks but to improve the classification performance. We found that creating adversarial samples is an effective way of data augmentation that addresses weak points of the currently trained CNN. In Sec. V we compare the efficiency of adversarial training with typical data augmentation.

V. Experiments

A. Experimental Setup

In this section, we report distress detection results that can be achieved on the GAPs dataset using different network architectures and techniques known from the literature to improve network performance. All networks have been trained on the training dataset of the 50k subset for 80 epochs, what has been verified to be a reasonable number of epochs on the validation dataset. We opted for the 50k subset as default dataset since the results on the smaller subset are comparable to the full dataset and even outperform the networks trained on the full dataset if data augmentation is applied (see Sec. V-D). We have used SGD with a momentum of 0.9 as the optimizer and a batch size of 64. The learning rate was varied in five steps between 0.005 and 0.08 (0.005, 0.01, 0.02, 0.04, 0.08) and every single experiment was repeated three times in a row. We computed F_1 score and Balanced Error Rate (BER) for every experiment, which is in compliance with [1] where these measures have been found to be most useful to measure performance on the GAPs dataset. Furthermore, we also report G-mean (GME) values to address the imbalanced class distribution of the dataset. Therefore, mean values over the trials of the best learning rate are reported. Standard deviation is given when sensible, but usually omitted since deviation for the best learning rate is low. Most experiments are carried out using a ResNet with 34 convolutional layers (ResNet34) since this model has been proven to fit well on the data. However, the effects of different network architectures have been analyzed in Sec. V-G.

B. Patch Size

We first analyzed if the neural network can benefit from additional context. Therefore, we have trained a ResNet34 on different patch sizes of the dataset. As shown in Fig. 5, the network greatly benefits from increasing the patch size, since the F_1 score improves significantly up to a patch size of 160×160 pixels. Therefore, it becomes evident that the suggested patch size of 64×64 in [1] hampers detection performance and should no longer be favored.

However, it becomes also obvious that the patch size does not increase much on the valid-test dataset with larger patch sizes. In fact, it even starts to decrease slightly on the test dataset.

TABLE III: Result obtained using a ResNet34 and a patch size of 160×160 pixel on the evaluation subsets of the GAPs 50k dataset. ↓ highlights that a lower score is better, whereas ↑ highlights that a higher score is better.

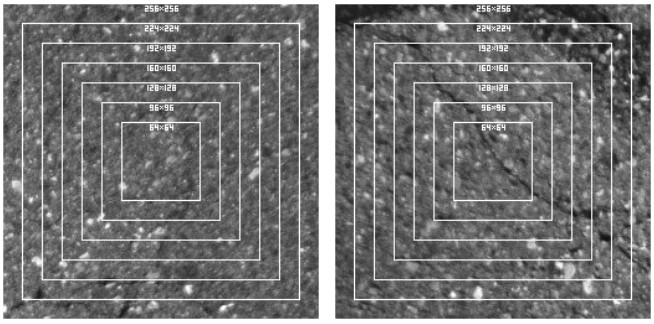

Fig. 4: Visualization of context captured by the different patch sizes

	valid			valid-test			test		
algorithm	F_1 ↑	BER ↓	GME ↑	F_1 ↑	BER ↓	GME ↑	F_1 ↑	BER ↓	GME ↑
ResNet34 $_{160\times160}$	.9284	.0592	.9408	.8982	.0857	.9141	.8709	.1090	.8902

As network inference performance decreases with the patch size (See fig. 5), we have decided in favor of a patch size of 160×160 pixels as it reflects a good trade-off between detection quality and inference speed. Therefore, the results on all subsets of the dataset are given in Tab. III.

C. Data Augmentation

Data augmentation often is used if the training dataset is of small size and is not diverse enough. Although the full gaps dataset is of a decent size, the 50k subset may not cover the same variety. Therefore, we want to show how data augmentation can improve the performance on the 50k dataset and are comparing the results to the full dataset in the next section. Since the pavement is captured from a bird-eye view in the GAPs dataset, we decided to randomly add rotation and a bit of translation to the patches in the training phase. The rotation was perturbed in the range of $-90°$ to $90°$, and a small translation offset of up to 5 percent of the patch size was added.

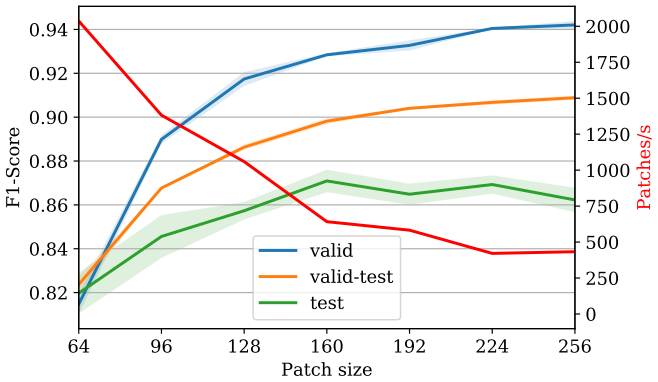

Fig. 5: Performance of ResNet34 trained on different patch sizes. The green line shows the inference performance reached with the different patch sizes on a NVIDIA TitanX graphics card.

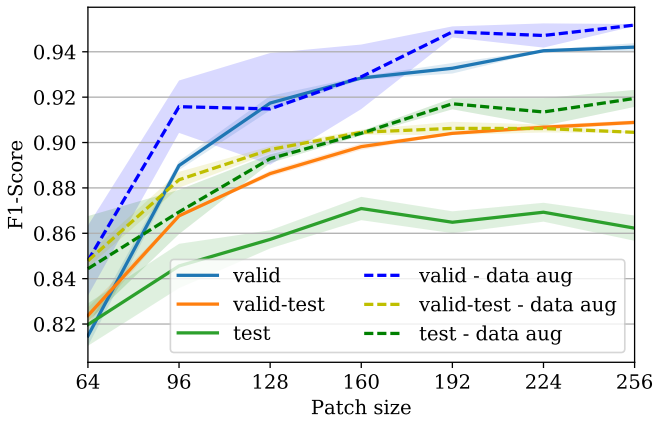

Fig. 6: Performance of ResNet34 trained on different patch sizes and with and without data augmentation on GAPs 50k. Results are very close and only a slight benefit can be observed on the valid-test subset if data augmentation is used.

The results for different patch sizes are given in Fig. 6. Although the performance increases for small patch sizes, the effect almost vanishes for larger patch sizes on the valid and valid-test dataset. Having a look at the test subset however (Fig. 6 and Tab. IV), discovers that the small improvement on the valid-test subset results in a much more significant improvement on the test subset. Therefore, neural networks can significantly benefit from data augmentation on the 50k GAPs dataset since it dramatically increases there generalization abilities.

TABLE IV: Effect of applying data augmentation to the 50k dataset. Results on test subset can be improved significantly (patch size 160×160, ResNet34).

	valid			valid-test			test		
algorithm	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑
No aug.	.9284	.0592	.9408	.8982	.0857	.9141	**.8709**	**.1090**	**.8902**
Data aug.	.9290	.0583	.9417	.9045	.0804	.9194	**.9041**	**.0820**	**.9174**

D. Full Dataset

We provide the GAPs dataset with a smaller 50k subset to enable researchers to test different networks and training techniques and to obtain results even with low computational resources. To show how the results obtained on the GAPs 50k dataset can be transferred to the full GAPs dataset, we compared both datasets by training a ResNet34 with patch size 128×128. While the patches of the full dataset are used unaltered, we have trained the model on the 50k dataset with the additional data augmentation described in V-C. The first thing that becomes obvious from the results of the training without data augmentation given in Tab. V is that the full dataset indeed offers a broader range of pavement characteristics. However, considering the difference in scale, the performance gap between the two datasets is not very huge. It even can be seen, that data augmentation can close the gap on the valid-test dataset and even performs better on the test dataset. That said, the 50k dataset seems to cover almost all different characteristics of damage and regular street, but adding rotation to the patches still increases variety.

TABLE V: Comparison between training on the GAPs 50k dataset with data augmentation and the full GAPs dataset.

	valid			valid-test			test		
algorithm	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑
50k	.9174	.0686	.9313	.8863	.0954	.9042	**.8573**	**.1201**	**.8792**
50k data aug.	.9148	.0718	.9280	.8969	.0868	.9128	**.8929**	**.0921**	**.9070**
Full DS	.9007	.0822	.9177	.8517	.1219	.8780	**.8667**	**.1106**	**.8892**

E. Transfer Learning

Transfer learning is known for producing excellent results across different domains. It has already been applied to the pavement distress detection domain and has lead to a significant performance boost [30]. Therefore, we have compared the training of a ResNet50 from scratch and a ResNet50 which weights have been pretrained on the ImageNet dataset.

Analyzing the training progress curves reveals that the best model on the valid-test dataset is obtained within 10 training epochs if transfer learning is used and if the learning rate is below 0.04. This is significantly faster than the model trained from scratch that reaches optimal performance after 40 epochs and indicates a good weight initialization. However, the transfer learning model reaches an even better performance with higher learning rates. For these learning rates, the faster learning process cannot be observed any longer.

Analyzes of the results obtained with the best models in Tab. VI reveal that there seems to be no real benefit from using transfer learning on the GAPs dataset. The scores on all the different subsets are almost the same and are even slightly better for the model trained from scratch.

TABLE VI: Comparison between ResNet50 trained from scratch and ResNet50 pretrained on ImageNet.

	valid			valid-test			test		
algorithm	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑
from scratch	.9375	.0511	.9489	.9004	.0839	.9157	**.8942**	**.0898**	**.9096**
transfer	.9324	.0561	.9438	.8970	.0863	.9135	**.8889**	**.0950**	**.9041**

F. Adversarial Training

Adversarial Training implicitly creates additional training samples and can be used as a goal-driven data augmentation. We applied this training technique while training networks on the 50k dataset and have also tested how the methods perform in combination with classical data augmentation (Sec. V-C). As a first result, adversarial training improves training stability as the results obtained are almost identical to all the different learning rates used in the experimental setup. As a second result, adversarial training hampers generalization on the test dataset if no traditional data augmentation is applied (See Tbl. VII). This is probably caused by the reduced diversity of the 50k dataset. If further data augmentation is applied, however, the results on the test dataset are catching up, leading to almost identical results with the traditional data augmentation approach. Since adversarial training requires additional training time for executing the backward and forward pass, the given

results do not justify the use of adversarial training on the 50k dataset.

TABLE VII: Effect of adversarial training with and without traditional data augmentation.

	valid			valid-test			test		
algorithm	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑
No aug.	.9284	.0592	.9408	.8982	.0857	.9141	**.8709**	**.1090**	**.8902**
Data aug.	.9290	.0583	.9417	.9045	.0804	.9194	**.9041**	**.0820**	**.9174**
Adverserial	.9256	.0615	.9385	.8981	.0853	.9145	**.8459**	**.1276**	**.8724**
Adv. + aug.	.9392	.0501	.9498	.9004	.0833	.9166	**.9023**	**.0837**	**.9157**

G. Network Architectures

The experiments with the ResNet34 and ResNet50 showed that the results obtained by both network architectures are very similar. Since inference time is crucial in the ASINVOS project, we tried to find out how much layers are needed to solve the detection problem with reasonable results. Therefore, we ran further experiments with a ResNet with 18 layers and created a ResNet with only convolutional 10 layers, that was also applied on the GAPs dataset (see Tbl. VIII). Furthermore, we have evaluated the dataset using the publicly available CrackIt toolbox [9] (Version 1.5 - 1-May-2016). However, as the error measures indicate, the algorithm was not able to perform well on the dataset using the parameter settings recommended by the authors, and the results are at random.

TABLE VIII: Comparison of Residual Networks with different depth. All experiments where carried out on patches of size 160×160 pixels of the GAPs 50k dataset using data augmentation.

	valid			valid-test			test		
algorithm	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑
ResNet10	.9467	.0457	.9542	.8965	.0864	.9135	**.8949**	**.0882**	**.9117**
ResNet18	.9413	.0483	.9517	.8979	.0852	.9147	**.9062**	**.0800**	**.9195**
ResNet34	.9290	.0583	.9417	.9045	.0804	.9194	**.9041**	**.0820**	**.9174**
ResNet50	.9375	.0511	.9489	.9004	.0839	.9157	**.8942**	**.0898**	**.9096**
CrackIt	.5712	.5000	.3530	.5714	.500	.4752	**.5714**	**.500**	**.3387**

Surprisingly, the 10 layer ResNet performed quite well considering its low complexity. However, although it performs very well on the validation dataset, its generalization abilities seem to suffer from the low number of layers. Together with the ResNet50 which, in contrast, suffers from its high number of layers, its performance on the test dataset is significantly lower compared to the 18 and 34 layer networks. Therefore, ResNets with 18 and 34 layers seem to perform well on the GAPs dataset. If inference time is of relevance, the ResNet18 probably is a good choice.

H. Multi-class Classification

The experiments in this paper are firmly focused on the distress detection problem. Nevertheless, we also present a baseline for the classification of all the different distress classes. Utilizing the results we have obtained on the distress detection problem, we have chosen a patch size of 160×160 pixels of the GAPs 50k dataset in combination with data augmentation. We have trained ResNets with 18 and 34 layers and also integrated test with class weighting to address the imbalanced class distribution. Therefore, weights have been chosen to resemble equal class distribution.

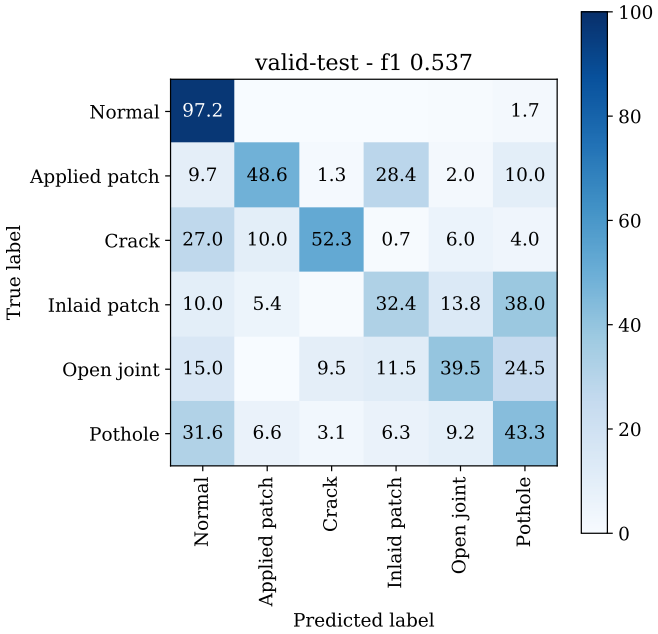

Fig. 7: Confusion matrix for the multi-class classification problem on the valid-test dataset using a ResNet18. Values below 1.0% are faded out for clarity.

As can be seen from Table IX, the classification performance drops significantly if the network should distinguish between the different distress classes. Although adding class weights or applying deeper networks seem to increase the performance on the test dataset, the classification results remain at a low level. This problem arises from the fact that it is sometimes quite hard to differentiate between certain distress classes. This is even true for experts of the field, and an additional frontal view is sometimes required to distinguish e.g. between an applied and an inlaid patch (see Fig. 7).

TABLE IX: Baseline results for multi-class classification with ResNet34 using data augmentation. We use the macro version of the F_1 score for multi-class evaluation.

	valid			valid-test			test		
algorithm	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑	$F_1 \uparrow$	BER ↓	GME ↑
ResNet18	.5678	.1957	.7527	.5368	.2115	.7764	**.4018**	**.3066**	**.6019**
ResNet18 Class weights	.5451	.2121	.7409	.5072	.2274	.7608	**.4445**	**.2894**	**.6324**
ResNet34	.5859	.1751	.7573	.5088	.2292	.7538	**.4635**	**.2859**	**.6464**

VI. Conclusion

We extended the GAPs dataset and improved the annotations. Furthermore, we created a CIFAR-like 50k subset that allows fast experimental evaluations regarding different neural network models, parametrization, and training techniques. Experiments showed that observations on this 50k subset generalize well to the full GAPs dataset.

TABLE X: Comparison between the ASINVOSnet proposed in [1] and the ResNet34 architecture. Results are obtained using data augmentation.

	valid			valid-test			test		
algorithm	$F_1 \uparrow$	BER $\downarrow$	GME $\uparrow$	$F_1 \uparrow$	BER $\downarrow$	GME $\uparrow$	$F_1 \uparrow$	BER $\downarrow$	GME $\uparrow$
ASINVOSnet 64×64	.8328	.1391	.8606	.7576	.2010	.7967	**.8131**	**.1566**	**.8391**
ResNet34 64×64	.8479	.1269	.8723	.8479	.1270	.8725	**.8444**	**.1318**	**.8664**
ASINVOSnet 160×160	.9311	.0574	.9425	.8752	.1035	.8964	**.8473**	**.1280**	**.8710**
ResNet34 160×160	.9290	.0583	.9417	.9045	.0804	.9194	**.9041**	**.0820**	**.9174**

Based on the 50k subset, we performed extensive experiments in order to identify possible performance gains in automated road condition assessment and were able to significantly improve detection performance compared to the network architecture presented in [1] (See Tbl. X). Our results show that

- more context improves the performance,
- data augmentation leads to better generalization,
- transfer learning does not help to improve the performance on the GAPs dataset,
- adversarial training may be useful to improve the stability against bad parametrization,
- the ResNet architecture should be favored over VGG-like models, and
- ResNets with 18 layers are capable of performing equally with deeper models.

While the results on distress detection are promising, the distinction of distress types still leaves space for future work.

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] F. Yang, "Feature pyramid and hierarchical boosting network for pavement crack detection," 2019.

[3] S. Anand, S. Gupta, V. Darbari, and S. Kohli, "Crack-pot: Autonomous road crack and pothole detection," *arXiv preprint arXiv:1810.05107*, 2018.

[4] H. Lee, "Application of machine vision techniques for the evaluation of highway pavements in unstructured environments," in *Proc. Fifth International Conference on Advanced Robotics Robots in Unstructured Environments*, 1991, pp. 1425–1428.

[5] T. Garbowski and T. Gajewski, "Semi-automatic inspection tool of pavement condition from three-dimensional profile scans," *Procedia Engineering*, vol. 172, pp. 310 – 318, 2017, modern Building Materials, Structures and Techniques.

[6] K. Gopalakrishnan, "Deep learning in data-driven pavement image analysis and automated distress detection: A review," *Data*, vol. 3, no. 3, p. 28, 2018.

[7] L. Peng, W. Chao, L. Shuangmiao, and F. Baocai, "Research on Crack Detection Method of Airport Runway Based on Twice-Threshold Segmentation," in *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*. IEEE, 2015, pp. 1716–1720.

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

[9] H. Oliveira and P. L. Correia, "CrackIT - An image processing toolbox for crack detection and characterization," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 798–802.

[10] W. Kaddah, M. Elbouz, Y. Ouerhani, V. Baltazart, M. Desthieux, and A. Alfalou, "Optimized minimal path selection (omps) method for automatic and unsupervised crack segmentation within two-dimensional pavement images," *The Visual Computer*, pp. 1–17, 2018.

[11] K. Fernandes and L. Ciobanu, "Pavement pathologies classification using graph-based features," *2014 IEEE International Conference on Image Processing, ICIP 2014*, pp. 793–797, 2014.

[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] T. Siriborvornratanakul, "An automatic road distress visual inspection system using an onboard in-car camera," *Advances in Multimedia*, vol. 2018, Article ID 2561953, 10 pages.

[14] R. Kapela, P. Sniatała, A. Błoch, and S. A. Atrem, "Asphalt Surfaced Pavement Cracks Detection Based on Histograms of Oriented Gradients," 2015.

[15] M. Quintana, J. Torres, and J. M. Menendez, "A Simplified Computer Vision System for Road Surface Inspection and Maintenance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 608–619, 2016.

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

[17] S. Chatterjee, P. Saeedfar, S. Tofangchi, and L. Kolbe, "Intelligent road maintenance: A machine learning approach for surface defect detection," in *Proceedings of the 26th European Conference onn Information Systems (ECIS), UK*, 2018, pp. 1–16.

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

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

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

[21] 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.

[22] S. Chatterjee, A. B. Brendel, and S. Lichtenberg, "Smart infrastructure monitoring: Development of a decision support system for vision-based road crack detection," in *Proceedings of the International Conference on Information Systems - Bridging the Internet of People, Data, and Things, (ICIS) 2018, San Francisco, CA, USA, December 13-16, 2018*, 2018.

[23] 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*, vol. 33, no. 12, pp. 1127–1141, 2018.

[24] K. Ma, M. Hoai, and D. Samaras, "Large-scale continual road inspection: Visual infrastructure assessment in the wild," in *Proceedings of British Machine Vision Conference*, 2017.

[25] Forschungsgesellschaft für Straßen- und Verkehrswesen, *ZTV ZEB-StB - Zusätzliche Technische Vertragsbedingungen und Richtlinien zur Zustandserfassung und -bewertung von Straßen [FGSV-Nr. 489]*. FGSV Verlag, 2006.

[26] A. Ng, "Nuts and bolts of building ai applications using deep learning." NIPS, 2016.

[27] 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.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[29] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Int. Conf. on Learning Representations (ICLR)*, 2015.

[30] M. Nie and K. Wang, "Pavement distress detection based on transfer learning," in *2018 5th International Conference on Systems and Informatics (ICSAI)*, 2018, pp. 435–439.