How to Get Pavement Distress Detection Ready for Deep Learning? A Systematic Approach

Markus Eisenbach [a], Ronny Stricker [a], Daniel Seichter [a], Karl Amende [a], Klaus Debes [a], Maximilian Sesselmann [b], Dirk Ebersbach [b], Ulrike Stoeckert [c], 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

[c] German Federal Road Research Institute (BASt)
51427 Bergisch Gladbach, Germany

markus.eisenbach@tu-ilmenau.de

***Abstract*—Road condition acquisition and assessment are the key to guarantee their permanent availability. In order to maintain a country's whole road network, millions of high-resolution images have to be analyzed annually. Currently, this requires cost and time excessive manual labor. We aim to automate this process to a high degree by applying deep neural networks. Such networks need a lot of data to be trained successfully, which are not publicly available at the moment. In this paper, we present the GAPs dataset, which is the first freely available pavement distress dataset of a size, large enough to train high-performing deep neural networks. It provides high quality images, recorded by a standardized process fulfilling German federal regulations, and detailed distress annotations. For the first time, this enables a fair comparison of research in this field. Furthermore, we present a first evaluation of the state of the art in pavement distress detection and an analysis of the effectiveness of state of the art regularization techniques on this dataset.**

I. Introduction

Public infrastructures are suffering from aging and therefore need frequent inspection. Distress detection and a solid management for maintenance are the key to guarantee their permanent availability. Therefore, condition acquisition and assessment must be applied to the whole road network of a country frequently. For Germany, this results in road surface condition acquisition of about 13,000 km freeways and about 40,000 km federal highways[1] with high-speed measurement vehicles and the assessment of the extracted data afterwards. Since the initial acquisition in 1991, the relevant surface characteristics are redetermined periodically in a four year cycle. Since freeways are inspected in both directions on all lanes, a total of approximately 30,000 lane-km need to be covered annually.

*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] German Federal Road Research Institute (Bundesanstalt für Straßenwesen BASt): Erfassen und Bewerten von Oberflächeneigenschaften, ZEB – Zustandserfassung und -bewertung von Straßen (Acquisition and assessment of road surface characteristics), info flyer, 2016.

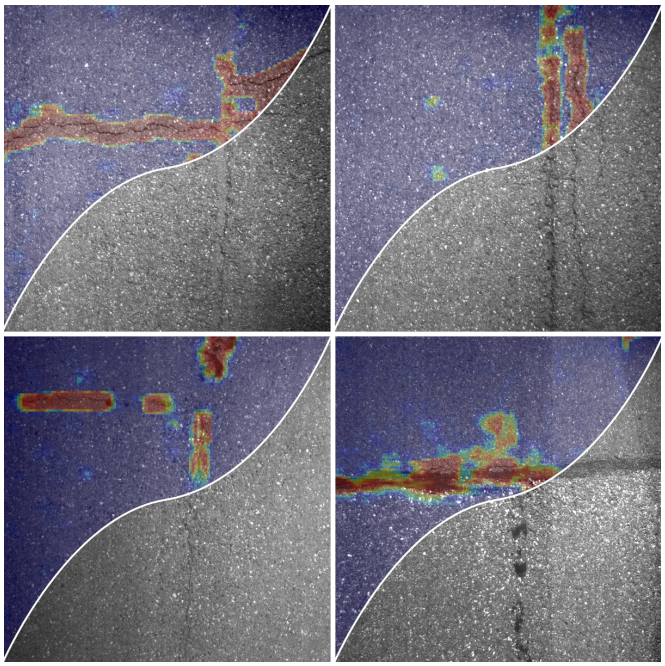


Fig. 1: Exemplary pavement images from the presented GAPs dataset with different types and levels of distress. Each image is overlayed with the output from a deep neural network for distress detection at the upper left part of the image. Red color suggests a high probability for distress, while blue symbolizes intact road.

Following German federal regulations, the surface characteristics have to be evaluated in terms of evenness, skid resistance, and substance condition. The surface characteristics evenness and skid resistance are mainly measured and analyzed automatically using laser sensors and sideway-force coefficient routine investigation machine technology (SCRIM). The substance condition is captured with camera systems

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

and has to be evaluated by visual inspection of the recorded images. Current evaluation is done manually and therefore requires excessive manual labor (evaluation of 2 000 images per lane-km). Therefore, the time span 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 for the population.

In the research project ASINVOS[2], 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 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 reduced, which helps to remedy deficiencies in time.

In this paper, we present the GAPs dataset, which is the first free and publicly available pavement distress dataset of a size, large enough to train high-performing deep neural networks. It provides high-quality images, recorded by a standardized process fulfilling German federal regulations, and detailed annotations. For the first time, this enables a fair comparison of alternative approaches in this field. Furthermore, we present a first evaluation of the state of the art in pavement distress detection on this dataset, followed by an analysis of the generalization ability of a deep neural network in the given road condition assessment domain. Therefore, we review the effectiveness of state of the art regularization techniques. Finally, we analyze, which performance measures should be reported in the future, to give a fair comparison.

II. Related Work

Automating the distress detection process has already attracted a lot of interest in the literature. Beside commercial all-in-one solutions like Dyntest, Pathway and the Applus System [1], whose internal algorithms are relatively unknown, a lot of different image processing approaches for automatic distress detection emerged in the literature during the last decade. The algorithms developed for evaluation of the pavement surface can be coarsely divided into three major groups: Crack image thresholding, patch-based classification, and depth-based algorithms.

A. Crack Image Thresholding

The first group of algorithms uses image processing methods to detect road distress structures that can be extracted by thresholding afterwards. Therefore, preprocessing algorithms are applied in order to reduce illumination artifacts. Under the assumption that crack structures can be identified as local intensity minima, thresholding in the image space is applied afterwards. The resulting crack image is further refined by morphological image operations and by searching for connected components. Approaches belonging to the aforementioned group are presented in [2], [3], [4], [5], as well as in [6], where the closed source but publicly available *CrackIT* toolbox is presented. This toolbox is included in the experimental evaluation of this paper. Other variants of that group use graph-based crack candidate analysis for further refinement [7], [8], [9], a *multi-scale curvelet* transform instead of a binary threshold [10], or gabor filters in order to find crack candidates [11].

B. Patch-based Classification

The algorithms of the second group apply different types of classifiers to patches of the image in order to extract crack or distress regions. Support vector machines (SVM) are commonly used. For example the classifier is applied to Histogram of Oriented Gradient (HOG) features [12] or Local Binary Patterns (LBP) [13], [14]. Neural networks are also applied in this domain, as for instance in [15], which describes an approach that uses a *Multi Layer Perceptron* network in combination with frequency features and image histograms. Other approaches rely on neural networks that do not require a separate feature extraction. For instance [16] use a *Multilayer Autoencoder*, and [17] use a *Convolutional Neural Network* for distress detection. The latter is included in the experimental evaluation of this paper.

C. Depth-based Algorithms

The third group of algorithms relies on depth information of the pavement. E.g. [18] proposes an algorithm for light section based crack detection, while [19] describes a method that relies on a 2D laser range finder. An algorithm that applies crack detection on 3D point clouds is given in [20]. Depth-based algorithms are excluded from our evaluation, since the presented dataset includes image data only.

D. Datasets

Although, a lot of different methods have been presented so far, there is a lack of publicly available datasets that are of descent size and are recorded in a standardized way. To our best knowledge, there are only three different datasets available, all of which have less than 300 images in total [5], [6], [7]. This hampers comparability, since most publications are using own datasets that have been generated using consumer cameras and are labeled in different ways.

III. GAPs Dataset

The German Asphalt Pavement Distress (GAPs) dataset[3] addresses the issue of comparability in the pavement distress domain by providing a standardized high-quality dataset of large size. This does not only enable researchers to compare their work with other approaches, but also allows to analyze algorithms on real world road surface evaluation data.

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

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

A. Standardized Data Acquisition

Accurate measurement data about the road's current condition are crucial for planning maintenance or expansion projects and reliable cost estimation. Thus, the German Road and Transportation Research Association (FGSV) developed a specific approach for collecting data of road condition – the so-called Road Monitoring and Assessment (RMA) [21]. The RMA process standardizes data acquisition on a systematic basis and provides nationwide uniform parameters to ensure objective analyses of surface conditions as well as a high degree of quality. The key aspects are longitudinal and transversal evenness, skid resistance and surface distresses. Mobile mapping systems, equipped with high-resolution cameras and laser-based sensors, are the state of the art in the RMA context.

1) Certification and quality standards: The German Federal Highway Research Institute (BASt) does not only participate in developing and optimizing such mobile mapping systems, but also acts as an approving authority for measurement systems and analysis processes that are deployed in the field of RMA. In Germany, providers of RMA services require an annual BASt certification to run RMA campaigns. This certification process includes static tests, like general technical checks of the measurement platform and its components, tests of the camera system using special test images, and tests of the laser sensors using a test specimen of granite. In addition, there are dynamic tests that include comparative measurements with the BASt reference measurement vehicles on a special proving ground (test against "golden device"). Apart from the data acquisition, the BASt certification process also includes strict reviews of the data analyses procedures.

2) Measurement vehicle: The data, that are presented in this paper, have been captured by the mobile mapping system S.T.I.E.R (Fig. 2). This measuring vehicle is manufactured and operated by the German engineering company LEHMANN + PARTNER GmbH. S.T.I.E.R has been designed for large-scale pavement condition surveys and is certified annually by the BASt since 2012. Therefore, it complies with the high German quality standards in the field of RMA. The main components of S.T.I.E.R are an inertial navigation system, laser sensors for evenness and texture measurements, a 2D laser range finder and different camera systems for capturing both the vehicles environment and the pavement's surface. The relevant data source for this paper is the surface camera system. It consists of two photogrammetrically calibrated JAI Pulnix TM2030 monochrome cameras. Each one features the Kodak KAI-2093 1" progressive scan CCD imager with $7.4\,\mu$m square pixels, a frame rate of 32 fps and a resolution of 1920×1080 pixels. 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 are mounted left and right at the rear of S.T.I.E.R's roof rack pointing at a right angle towards the road. As each camera image covers a pavement patch of $2.84\,\text{m}\times1.0\,\text{m}$, both images combined describe the entire driven lane.

Fig. 2: Mobile mapping system S.T.I.E.R

3) RMA-specified labeling: Within the scope of the conventional RMA workflow, a sequence of left and right surface camera images is stitched together in driving direction. The result is a continuous sequence of surface images that represent 10 meters of the entire driven traffic lane. According to the FGSV-regulation, the surface damage detection and analysis process is based on these images. For this, an inspection grid is applied to each 10-meter-image (see Fig. 3). A single grid cell has a longitudinal length of 1 m and a transversal length of $\frac{1}{3}$ of the lane width. If a grid cell contains a relevant surface damage, the whole cell is assigned to this damage type. Once the damage detection and classification is done, the measured raw-data is used to calculate condition variables and finally condition grades ranging from "very good" to "very poor"

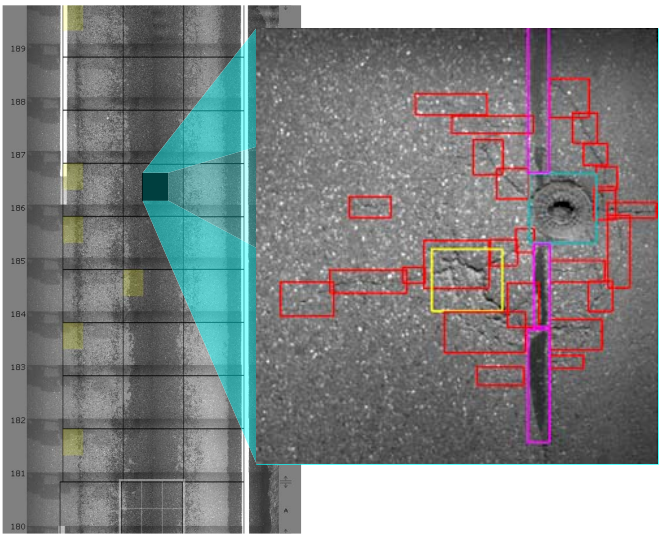


Fig. 3: left: Labeling as expected by German FGSV-regulation. right: fine labeling of different distress types using bounding boxes.

using a weighting scheme defined by the FGSV. The presented conventional labeling approach is sufficient for indicating the level of safety and comfort for road users, but due to the lack of the precise damage location labels in terms of pixel coordinates, this labeling is not appropriate to train a classifier. Also stitched images are problematic, since artificial edges at stitched image borders may complicate the learning process.

4) Dataset for neural network training: To provide a high-quality training dataset, we use the HD-images from the left and right surface camera instead of stitched images. The GAPs dataset includes a total of 1 969 gray valued images (8 bit), partitioned into 1 418 training images, 51 validation images, and 500 test images. The image resolution is 1920×1080 pixels with a per pixel resolution of 1.2 mm×1.2 mm. The pictured surface material contains pavement of three different German federal roads. Images of two German federal roads are used for training. Another section of one of these roads is used for validation. The two roads can be characterized by relatively poor pavement condition. The third German federal road is uniquely used for testing. Its condition is better. Thus the ratio of intact to defect road surface differs significantly from the other two roads. The data acquisition took place in summer 2015, so the measuring condition were dry and warm.

The images have been annotated manually by trained operators at a high-resolution scale (see Fig. 3) such that an 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. The relevant damage classes are cracks, potholes, inlaid patches, applied patches, open joints and bleedings (see Fig. 4). Cracks are the dominant damage class. This class comprises all sorts of cracks like single/multiple cracking, longitudinal/transversal cracking, alligator cracking, and sealed/filled cracks.

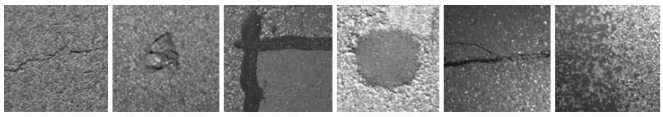


Fig. 4: Surface defect classes as defined by FGSV-regulation: CRACK – Crack*, POTHO – Pothole*, INPAT – Inlaid patch*, APPAT – Applied patch*, OPJOI – Open joint*, BLEED – Bleeding (not present in acquired images of GAPs dataset). *Class is included in GAPs dataset.

IV. Evaluated Algorithms

Comparing different methods for distress detection is currently hindered by the lack of a sophisticated and publicly available benchmark dataset. Therefore, we have selected two different state of the art methods that are evaluated in the experiments section of this paper.

A. Image Thresholding Approaches

We have used the publicly available CrackIT Toolbox [6] as representative for the image thresholding based approaches. The toolbox provides different algorithms for image preprocessing and crack detection based on pattern classification techniques. It provides an image preprocessing stage that includes filters for context aware image smoothing and a dedicated lane line detection module to remove lane lines from the input image. Furthermore, the toolbox applies local image block-based normalization to reduce illumination dependence. Assuming low intensity values for crack pixels, the image is thresholded afterwards by analyzing the standard deviation of the intensity values of local image blocks. The resulting binary crack candidates image is refined afterwards by a connected-component algorithm in order to identify relevant cracks pixels.

The results of the toolbox are very sensitive to changes in the parameters used for the different processing steps. Therefore, the authors suggest to tune the parameters for the desired field of application, which we did on the GAPs training data.

B. Deep Learning Approaches

As representative for deep learning approaches, we have decided in favor of the promising *Convolutional Neural Network* (CNN) approach [17] for road crack detection (in the following referred to as RCD net). Zhu *et al.*[17] presented a relatively small CNN with four blocks with alternating convolutional and max-pooling layers and two fully-connected layers (see Fig. 5), inspired by LeNet [22] architecture.

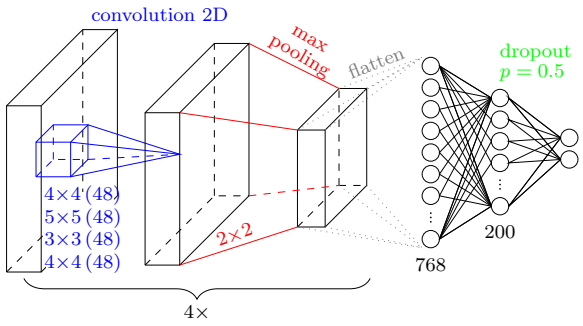


Fig. 5: Structure of RCD net [17]

TABLE I: RCD model [17]. Abbreviations: D – dropout (dropout probability), W – weight decay, in – input, conv – convolution, pool – max pooling, fc – fully connected layer, out – softmax output

type	filter size	stride	regular.	output size	# paramet.
in				3×99×99	
conv	4×4 (48)	1×1	W (0.0005)	48×96×96	2 352
pool	2×2	2×2		48×48×48	
conv	5×5 (48)	1×1	W (0.0005)	48×44×44	57 648
pool	2×2	2×2		48×22×22	
conv	3×3 (48)	1×1	W (0.0005)	48×20×20	20 784
pool	2×2	2×2		48×10×10	
conv	4×4 (48)	1×1	W (0.0005)	48×7×7	36 912
pool	2×2	2×2		48×4×4	
flat				768	
fc	(200)		W (0.0005)	200	153 800
out	(2)		D (0.5), W (0.0005)	2	402
					271 898

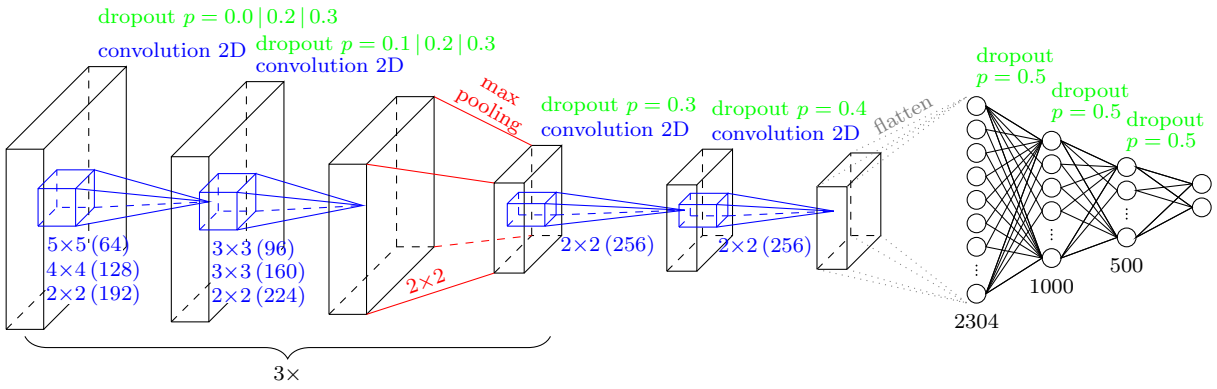


Fig. 6: Structure of ASINVOS net

By using only small filter sizes for the convolutional layers and only 48 filters per layer, the net is very compact and has only 0.3 M weights (see Tab. I). Therefore, this network does not need much regularization. Only weight decay and dropout for the last fully-connected layer is used.

The authors provided us their CAFFE [23] code and their dataset, so we could check equality of results. We got a comparable but slightly better results on their data (the exact partitioning of their dataset could not be recovered). We then reimplemented the RCD using Keras [24] based on Theano [25] and integrated it into our framework. By parameter tuning, again, we could slightly improve the net's performance. For the best parameter setting see Tab. IV.

Since the RCD net is relatively small and does not represent modern CNN architectures, that are deeper, we conceptualize another CNN with eight convolutional layers, three max-pooling layers, and three fully connected layers (referred to as ASINVOS net in the following), and implemented it using Keras [24] based on Theano [25]. Its architecture is inspired by the ImageNet winning VGG-models [26] (multiple units of two convolutional layers followed by one max-pooling layer) and AlexNet [27] (fully connected layers with softmax output). All neurons are ReLUs [28]. For the exact architecture, with filter sizes and dropout rates see Fig. 6.

The ASINVOS net has 4.0 M weights (see Tab. II). Thus, regularization is the key to perform well on unknown data.

1) Regularization: Dropout [29] is known to be a very good regularization technique that avoids co-adaption and also improves generalization abilities. Therefore, our first approach is the extensive use of dropout for all layers except the input layer as sole regularization technique. Recently, this has been successfully applied in the person detection domain [30].

In recent years, batch normalization [31] replaces dropout in modern neural network architectures. It is well known, that input normalization (zero-mean, unit variance) as a pre-processing step can improve neural network training. Batch normalization takes this idea even further and aims to remove the covariate shift from the internal activation of each subsequent layer. Thus, batch normalization can speed up training and often leads to a higher accuracy. We pursue two approaches: First, we replace dropout with batch normalization. Second, we use dropout in combination with batch normalization.

TABLE II: ASINVOS model. Abbreviations as in Tab. I.

type	filter size	stride	regular.	output size	# paramet.
in				1×64×64	
conv	5×5 (64)	1×1	—	64×60×60	1 664
conv	3×3 (96)	1×1	D (0.1)	96×58×58	55 392
pool	2×2	2×2		96×29×29	
conv	4×4 (128)	1×1	D (0.2)	128×26×26	196 736
conv	3×3 (160)	1×1	D (0.2)	160×24×24	184 480
pool	2×2	2×2		160×12×12	
conv	2×2 (192)	1×1	D (0.3)	192×11×11	123 072
conv	2×2 (224)	1×1	D (0.3)	224×10×10	172 256
pool	2×2	2×2		224×5×5	
conv	2×2 (256)	1×1	D (0.3)	256×4×4	229 632
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
					4 032 134

Since large weights may impair generalization abilities of a neural network, penalizing them is considered as a good regularization mechanism. We evaluated two approaches, namely weight decay [32] and max-norm regularization [29]. To find appropriate hyper-parameters, we plotted the norm of weights in different layers over epochs. Once the network started to overfit, we determined the norms at this epoch and derived suitable hyper parameters.

2) Network Structure Variation: To evaluate, if the network structure can be improved, we set up two experiments: First, we evaluate the input coding. We recognized that the gray value histogram showed a distribution composed of three normal distributions (road paint, asphalt color, and shadows due to pavement structure). Therefore, we chose a topological input coding with three neurons per pixel.

Second, based on findings in [33], that each convolutional filter larger than 3×3 can be replaced by multiple 3×3-filters, we rearranged our structure. 5×5-filters are replaced by two successive 3×3-filters and the 4×4-filter by a 3×3-filter. Szegedy *et al.* [33] also found, that the performance

improves when two successive 2×2-filters are replaced by one 3×3-filter. With this replacement, the modified ASINVOS net (referred to as ASINVOS-mod) used 3×3-filters only. Based on modern neural network architectures, e.g. [33], [31], [34], where pooling layers are replaced by filter map reducing convolutional layers with a stride of 2×2, we also replaced all pooling layers by such convolutional layers, that learn the reduction. We also followed these state of the art nets by replacing valid convolutions with size preserving convolutions by the use of zero-padding. The modified structure is shown in Tab. III. The increased number of layers and the use of size preserving convolutions result in an increase of parameters to 18.3 M. This increases the explanatory power of the neural network, but may negatively affect the generalization abilities.

TABLE III: ASINVOS-mod model. Abbreviations as in Tab. I.

type	filter size	stride	regular.	output size	# paramet.
in				1×64×64	
conv	3×3 (64)	1×1	—	64×64×64	640
conv	3×3 (64)	1×1	D (0.1)	64×64×64	36 928
conv	3×3 (96)	1×1	D (0.1)	96×64×64	55 392
conv	2×2 (96)	2×2	D (0.1)	96×32×32	36 960
conv	3×3 (128)	1×1	D (0.2)	128×32×32	110 720
conv	3×3 (160)	1×1	D (0.2)	160×32×32	184 480
conv	2×2 (160)	2×2	D (0.2)	160×16×16	102 560
conv	3×3 (192)	1×1	D (0.3)	224×16×16	276 672
conv	2×2 (192)	2×2	D (0.3)	192×8×8	147 648
conv	3×3 (256)	1×1	D (0.3)	256×8×8	442 624
flat				16384	
fc	(1000)		D (0.5)	1000	16 385 000
fc	(500)		D (0.5)	500	500 500
out	(2)		D (0.5)	2	1 002
					18 282 278

TABLE IV: CNN model comparison

approach	RCD [17]	ASINVOS	ASINVOS-mod
neurons			
neuron type	ReLU	ReLU	ReLU
model size			
input	99×99 RGB	64×64 gray	64×64 gray
depth	6 layers	11 layers	13 layers
# weights	0.3 M	4.0 M	18.3 M
layer configuration			
conv filter size	3×3 – 5×5	2×2 – 5×5	3×3
feature map reduction	max-pooling stride 2×2	max-pooling stride 2×2	convolution stride 2×2
regularization			
dropout	1×($p = 0.5$)	10×(0.1 – 0.5)	12×(0.1 – 0.5)
weight decay	0.0005	—	—
learning parameters			
optimizer	SGD	SGD	SGD
learning rate	0.001	0.01	0.01
momentum	0.9	0.7	0.9
batch size	48	256	256

V. Evaluation of State of the Art on GAPs Dataset

To show the capabilities of deep neural networks for distress detection on road surfaces, we applied them on the presented GAPs dataset (see Fig. 1 for visual results). Furthermore, we compare the results with classical computer vision algorithms that are common in the research community.

A. Evaluation Protocol

To evaluate the pure classification performance of the different algorithms instead of the detection performance, we extracted image patches. Using a sampling strategy that favors distress over intact road, we extracted a total of 4.9 M patches for training (approximately $\frac{1}{8}$ distress), 200 k patches for validation ($\frac{1}{4}$ distress) and 1.2 M patches for testing purpose (better pavement condition, only $\frac{1}{20}$ distress). All algorithms are evaluated on patches of size 64×64, except the RCD net that is additionally evaluated on 99×99-patches (drawn at the same positions), since this is its intended input size.

The dataset contains different types of road paint (arrows, lane lines, etc.). Since first evaluations with the CrackIT toolbox revealed, that it fails to handle road paint robustly, we excluded patches with road paint from the evaluation. We consider this a fairer comparison.

B. Performance Measures

To make future comparisons as easy as possible, we report all common performance measures, that can be derived from the Receiver Operator Characteristic (ROC) curve and the Precision Recall (PR) curve. The full listing with equations and used abbreviations is shown in Tab. V.

TABLE V: Performance Measures, where tp is the number of true positives, fn the number of false negatives, fp the number of false positives, and tn the number of true negatives.

measure	abbreviation, equation	curve
true positive rate, Recall	$\text{TPR} = \text{REC} = \frac{tp}{tp+fn}$	ROC, PR
true negative rate	$\text{TNR} = \frac{tn}{fp+tn}$	ROC
precision	$\text{PRC} = \frac{tp}{tp+fp}$	PR
balanced error rate	$\text{BER} = 1 - \frac{1}{2}(\text{TPR} + \text{TNR})$	ROC
accuracy	$\text{ACC} = \frac{tp+tn}{tp+fn+fp+tn}$	ROC
Matthews correlation coefficient	$\text{MCC} = \frac{tp \cdot tn - fp \cdot fn}{\sqrt{\alpha}}$ $\alpha = (tp+fp)(tp+fn)(tn+fp)(tn+fn)$	ROC
G-mean	$\text{GME} = \sqrt{\text{TPR} \cdot \text{TNR}}$	ROC
area under ROC curve	AUC	ROC
F_1 score	$\text{F}_1 = 2 \cdot \frac{\text{PRC} \cdot \text{REC}}{\text{PRC} + \text{REC}}$	PR
G-measure	$\text{GMS} = \sqrt{\text{PRC} \cdot \text{REC}}$	PR
break even point	$\text{BEP} = \text{PRC} \overset{!}{=} \text{REC}$	PR
area under precision recall curve	APR	PR

We favor two measures: The F_1 score, that is derived from the PR curve, and the balanced error rate (BER), that is derived from the ROC curve. The often reported accuracy (ACC) is not a good performance measure when dealing with unbalanced real-world data (as e.g. the GAPs dataset), since it favors the dominant class. This is usually the negative class (in our case intact road), which is not the primary focus of detection. Also area under ROC or PR curve is not a favorable measure, since many irrelevant parts of the curve (high errors or low detection rates) have a large influence on the measure.

If we should choose between F_1 score and balanced error rate (BER), we would decide in favor of the F_1 score, since

it clearly focuses on the positive class (distress) and reports a combination of detection rate (recall) and false alarm rate (precision). These two characteristics are most valuable to rate the applicability of a real-world detector.

C. Computational Effort

Experiments for regularization, network structure, and all the other parameter evaluations ran for approximately three months on two NVIDIA Titan X GPUs using Keras [24] with Theano [25] backend. The average time to train a model on a single GPU was 10 days. In the execution phase, the processing of an HD image takes less than a half second.

D. Regularization Evaluation

To be able to evaluate the generalization abilities of deep learning approaches, the GAPs dataset is partitioned such that the validation data is more similar to the training data than the test data. This was achieved by extracting the validation data from one of the roads that was also used for training, but from a different section of that road. In contrast, the test data are extracted from another German federal road with completely other road surface conditions (less distress).

Tab. VIII shows the validation results, while Tab. IX shows the test results (table sections *original* and *regularization*). It can be seen, that the performance decreases from validation to test for all analyzed networks (original and all regularization). We conclude, that the test data differ more from training data than the validation data and thus, cause worse results. The generalization abilities are not sufficient to cope with unknown and significantly different data. None of the regularization techniques can cause a substantial improvement over the results achieved with dropout in the original ASINVOS net.

According to the balanced error rate (BER), batch normalization without dropout achieved the best test results (Tab. IX), closely followed by dropout only (ASINVOS net). Dropout in combination with batch normalization performs worst. Penalizing large weights with weight decay or max-norm regularization decreases the performance.

If the F_1 score is used to rate the performance (which we prefer), using dropout as sole regularization technique is the best choice, followed by batch normalization only. Combining both approaches is unfavorably. Again, both weight decay and max-norm regularization decreased the performance.

As a result, we propose to use either dropout or batch normalization for all layers, but neither weight decay nor max-norm for regularization.

E. Network Structure Evaluation

Tab. VIII (validation results) and Tab. IX (test results) (table sections *original* and *network structure*) show, that the chosen topological coding clearly performs worse than the pure gray value input coding (indicated by all performance measures).

We conclude that a CNN is able to learn the input coding by its own. Furthermore, an increase of input dimensions without an increase of information can decrease the performance significantly. Thus, it should be avoided.

An adaption of the network structure substantially improved the validation result (indicated by all performance measures but APR), but the generalization abilities have dropped. For the completely different road in the test data the net with the modified structure (ASINVOS-mod) performs worse than the original ASINVOS net (indicated by all performance measures but AUC) due to the increased number of weights. Therefore, we have a mixed result.

The modifications are promising, but to achieve better results on unknown and different data, the regularization must be improved.

In future work, we will also evaluate more advanced network structures like ResNets [34], [35], latest InceptionNets [33], [36], and sequential classifiers such as stack CNN-RNN [37], [38].

F. Distress Type Detection Analysis

To ensure, the networks did not only learn to detect the dominant distress class of cracks, but are able to recognize each distress class appropriately, we analyzed the detection results of the ASINVOS and the ASINVOS-mod net in detail. The threshold parameters were chosen such that the F_1 score had a peak in the precision recall curve and produced the best result as listed in Tab. VIII and IX repectively.

Tab. VI shows, that both classifiers can robustly detect each type of distress. Also on the test set (Tab. VII) they perform well.

We conclude, that the presented CNNs are able to learn appropriate features to recognize all types of distress.

TABLE VI: GAPs dataset validation results per surface distress class. Abbreviations as in Fig. 4.

	ASINVOS net		ASINVOS-mod net	
type	detected	%	detected	%
CRACK	39110/43065	90.82	39344/43065	91.36
POTHO	4420/4496	98.31	4436/4496	98.67
INPAT	662/746	88.74	688/746	92.23
APPAT	1037/1186	87.44	1059/1186	89.29
OPJOI	2048/2098	97.62	2078/2098	99.05
BLEED	0/0	—	0/0	—

TABLE VII: GAPs dataset test results per surface distress class. Abbreviations as in Fig. 4.

	ASINVOS net		ASINVOS-mod net	
type	detected	%	detected	%
CRACK	18665/23799	78.43	19763/23799	83.04
POTHO	62/68	91.18	60/68	88.24
INPAT	340/391	86.96	381/391	97.44
APPAT	10243/11830	86.58	9913/11830	83.80
OPJOI	649/674	96.29	659/674	97.77
BLEED	0/0	—	0/0	—

G. Comparison to State of the Art

For a comparison of the state of the art see test results in Tab. IX (ASINVOS net, ASINVOS-mod, RCD net, CrackIT). Analyzing the results, it is quite obvious, that deep learning approaches (ASINVOS net, ASINVOS-mod, RCD net) clearly outperform classical computer vision methods like CrackIT.

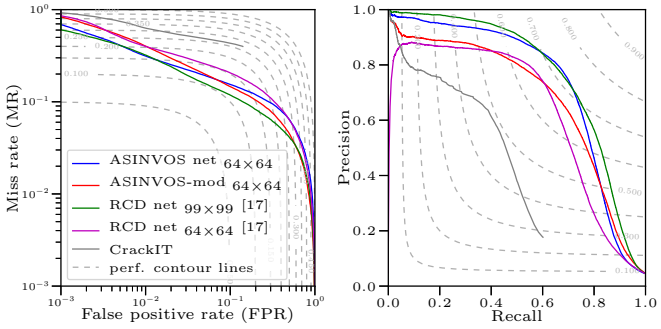
TABLE VIII: ASINVOS dataset validation results. Abbreviations as in Tab. V. Arrows beside the performance measures show if greater (↑) or lower (↓) results are better. The best result achieved for each performance measure is highlighted in bold. All performance measures are chosen at the working points of their peak in the respective curve (ROC / PR). TPR and TNR are chosen at the working point where BER peaks. The preferred performance measures are highlighted in blue.

algorithm	validation										
	TPR ↑	TNR ↑	BER ↓	ACC ↑	MCC ↑	GME ↑	AUC ↑	F_1 ↑	GMS ↑	BEP ↑	APR ↑
original											
ASINVOS net $_{64\times64}$	0.9164	0.9411	0.07124	0.9431	0.8493	0.9287	0.9758	0.8876	0.8876	0.8871	**0.9482**
regularization											
+Batch Normalization	0.9068	0.9421	0.07551	0.9422	0.8463	0.9243	0.9711	0.8843	0.8846	0.8828	0.9437
+Batch Norm –Dropout	0.9026	0.9310	0.08317	0.9348	0.8264	0.9167	0.9684	0.8698	0.8699	0.8684	0.9343
+Weight Decay $_{(0.0002)}$	0.8702	0.9614	0.07709	0.9386	0.8369	0.9229	0.9717	0.8780	0.8780	0.8772	0.9394
+Max-norm $_{(1.45)}$	0.9060	0.9382	0.07790	0.9387	0.8375	0.9220	0.9710	0.8784	0.8785	0.8776	0.9410
network structure											
ASINVOS-mod $_{64\times64}$	0.9227	0.9466	**0.06518**	**0.9480**	**0.8624**	**0.9347**	**0.9777**	**0.8973**	**0.8973**	**0.8970**	0.9475
Topo coding $_{3\times64\times64}$	0.9073	0.9383	0.07713	0.9390	0.8375	0.9228	0.9708	0.8785	0.8785	0.8782	0.9414
state of the art											
RCD net $_{64\times64}$ [17]	0.8105	0.8646	0.16240	0.8759	0.6628	0.8371	0.9099	0.7470	0.7471	0.7467	0.7982
RCD net $_{99\times99}$ [17]	0.9194	0.9429	0.06882	0.9473	0.8478	0.9311	0.9758	0.8821	0.8821	0.8820	0.9395
CrackIt	0.6182	0.8832	0.24930	0.7596	0.5390	0.7400	0.7982	0.7123	0.7139	0.7093	0.8194

TABLE IX: ASINVOS dataset test results. Abbreviations as in Tab. V and highlighting as in Tab. VIII. For a visualization of the performance curves in the detection error tradeoff (DET) diagram, that is derived from the ROC curve, and the precision recall (PR) diagram, see Fig. 7

algorithm	test										
	TPR ↑	TNR ↑	BER ↓	ACC ↑	MCC ↑	GME ↑	AUC ↑	F_1 ↑	GMS ↑	BEP ↑	APR ↑
original											
ASINVOS net $_{64\times64}$	0.8149	0.9432	0.1209	**0.9772**	**0.7148**	0.8771	0.9221	**0.7246**	**0.7266**	**0.7158**	0.7367
regularization											
+Batch Normalization	0.7977	0.9433	0.1294	0.9741	0.6730	0.8685	0.9132	0.6857	0.6868	0.6832	0.7062
+Batch Norm –Dropout	0.8417	0.9213	0.1184	0.9741	0.6690	0.8808	0.9378	0.6813	0.6826	0.6784	0.7112
+Weight Decay $_{(0.0002)}$	0.8046	0.9238	0.1358	0.9737	0.6541	0.8624	0.9007	0.6660	0.6683	0.6608	0.6909
+Max-norm $_{(1.45)}$	0.8179	0.9385	0.1218	0.9759	0.6911	0.8761	0.9233	0.7012	0.7036	0.6949	0.7273
network structure											
ASINVOS-mod $_{64\times64}$	0.8372	0.9206	0.1211	0.9723	0.6561	0.8780	0.9322	0.6707	0.6711	0.6698	0.6690
Topo coding $_{3\times64\times64}$	0.8042	0.9168	0.1395	0.9723	0.6413	0.8588	0.9075	0.6518	0.6555	0.6440	0.6641
state of the art											
RCD net $_{64\times64}$ [17]	0.7757	0.9221	0.1511	0.9732	0.6541	0.8457	0.9095	0.6642	0.6676	0.6500	0.6339
RCD net $_{99\times99}$ [17]	0.8553	0.9389	**0.1029**	0.9769	0.7077	**0.8962**	**0.9430**	0.7184	0.7199	0.7152	**0.7684**
CrackIt	0.5394	0.9315	0.2645	0.9607	0.4779	0.7188	0.7601	0.4882	0.4969	0.4694	0.4251

Furthermore, CrackIT is extremely sensitive to the chosen parameters, leading to bad generalization, as can be seen by the performance drop between similar data (validation, Tab. VIII) and significantly different data (test, Tab. IX).



(a) DET curve (b) Precision-Recall

Fig. 7: (a) Detection error tradeoff (DET) curve and (b) precision recall diagram on GAPs test data. Dashed lines show the contour plots of the derived performance measures (a) balanced error rate (BER) and (b) F_1 score (see Tab. IX).

Surprisingly, the network with significantly more weights (ASINVOS net) performed only slightly better that the relatively small RCD net $_{99\times99}$ [17]. However, when comparing results for equal input sizes of 64×64 pixels, the ASINVOS net outperforms the RCD net by far. We conclude, that 64×64-patches do not provide enough context information. Thus larger input patches should be the focus of future evaluations.

VI. Conclusion

Since road condition acquisition and assessment is important to maintain a country's road network, millions of high-resolution road surface images are analyzed annually. In order to replace the extensive manual labor by an automatic distress detection system with high-performing deep neural networks, much data for training is needed. Therefore, we presented the GAPs dataset, which is the first freely available pavement distress dataset of a size, large enough to train modern deep neural networks. The dataset images are recorded by a standardized process fulfilling German federal regulations. For each image, detailed distress annotations are available.

Thus, for the first time we were able to evaluate the state of the art in pavement distress detection in a meaningful way and reporting appropriate performance measures, namely balanced error rate (BER) and F_1 score. Summarized, only deep learning approaches were able to achieve satisfying detection results. Conventional computer vision approaches were beaten by a large margin. Furthermore, we analyzed the effectiveness of state of the art regularization techniques including dropout [29], batch normalization [31], max-norm regularization [29] and weight decay [32]. The best generalization results were achieved using dropout only, followed by batch normalization only. Penalizing large weights decreased the performance.

With the presented dataset and the extensive evaluation of deep neural networks, we made a first step to automate the time and labor intensive process of analyzing millions of road surface images annually.

REFERENCES

[1] J. Laurent, J. F. Hébert, D. Lefebvre, and Y. Savard, "Using 3d laser profiling sensors for the automated measurement of road surface conditions," in *RILEM Int. Conf. on Cracking in Pavements*. Springer, 2012, pp. 157–167.

[2] W. Huang and N. Zhang, "A novel road crack detection and identification method using digital image processing techniques," in *Int. Conf. on Computing and Convergence Technology (ICCCT)*. IEEE, 2012, pp. 397–400.

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] K. Fernandes and L. Ciobanu, "Pavement pathologies classification using graph-based features," in *Int. Conf. on Image Processing (ICIP)*. IEEE, 2014, pp. 793–797.

[10] 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 (ICIA)*. IEEE, 2015, pp. 647–652.

[11] 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.

[12] 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*, 2015, pp. 579–584.

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

[14] 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.

[15] H. Zakeri, F. M. Nejad, A. Fahimifar, A. D. Torshizi, and M. F. Zarandi, "A multi-stage expert system for classification of pavement cracking," in *Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*. IEEE, 2013, pp. 1125–1130.

[16] L. Shi, C. Gao, and J. Zhang, "Pavement distress image recognition based on multilayer autoencoders," in *Int. Conf. on Artificial Intelligence and Computational Intelligence (AICI)*. Springer, 2012, pp. 666–673.

[17] 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.

[18] C. Mertz, "Continuous road damage detection using regular service vehicles," in *ITS World Congress*, 2011.

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

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

[21] 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.

[22] 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.

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[24] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

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

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[32] J. Moody, S. Hanson, A. Krogh, and J. A. Hertz, "A simple weight decay can improve generalization," *Advances in neural information processing systems*, vol. 4, pp. 950–957, 1995.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conf. on Computer Vision (ECCV)*, 2016, pp. 630–645.

[36] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

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

[38] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional mkl based multimodal emotion recognition and sentiment analysis," in *Int. Conf. on Data Mining (ICDM)*. IEEE, 2016, pp. 439–448.