data

MDPI

Review

Deep Learning in Data-Driven Pavement Image Analysis and Automated Distress Detection: A Review

Kasthurirangan Gopalakrishnan

Department of Civil, Construction, and Environmental Engineering, Iowa State University, Ames, IA 50011, USA; rangan@iastate.edu

Received: 15 June 2018; Accepted: 18 July 2018; Published: 24 July 2018

Abstract: Deep learning, more specifically deep convolutional neural networks, is fast becoming a popular choice for computer vision-based automated pavement distress detection. While pavement image analysis has been extensively researched over the past three decades or so, recent ground-breaking achievements of deep learning algorithms in the areas of machine translation, speech recognition, and computer vision has sparked interest in the application of deep learning to automated detection of distresses in pavement images. This paper provides a narrative review of recently published studies in this field, highlighting the current achievements and challenges. A comparison of the deep learning software frameworks, network architecture, hyper-parameters employed by each study, and crack detection performance is provided, which is expected to provide a good foundation for driving further research on this important topic in the context of smart pavement or asset management systems. The review concludes with potential avenues for future research; especially in the application of deep learning to not only detect, but also characterize the type, extent, and severity of distresses from 2D and 3D pavement images.

Keywords: pavement cracking; pavement management; pavement imaging; 3D image; deep learning; TensorFlow; deep convolutional neural networks

1. Introduction

Transportation infrastructure systems are essential to the minimum operations of the government and commerce, and are considered the backbone of a nation's economy. Yet, they are literally crumbling across the globe and are considered "to be on life support" even by authorities in charge of maintaining them, especially in the United States. According to the 2017 American Society of Civil Engineers (ASCE) Infrastructure Report Card, the road infrastructure in the United States received a 'D' grade [1]. This state is due, in large part, to delayed maintenance and underinvestment in upgrades of transportation infrastructure systems. This, combined with increasing budgetary constraints, necessitates the development of efficient structural and functional health monitoring techniques for early detection of distresses developing in pavements. This can lead to significant cost savings resulting from timely maintenance and repair activities.

Highway agencies typically employ dedicated pavement data collection vehicles equipped with high-speed digital cameras or 3D laser scanner for inspecting the pavement surface, and acquire 2D or 3D pavement images [2,3]. In recent years, the 3D automated survey systems have also been introduced to acquire high-resolution 3D images of the pavement surface, which also offers opportunities for detecting other distresses apart from cracking [4,5].

Automated detection of distresses from pavement images (see Figure 1) is a challenging problem that has been quite thoroughly studied by the computer vision research community for more than three

decades. However, the challenges associated with 2D pavement images, such as variations in image source (digital camera, smartphone, unmanned aerial vehicle (UAV), etc.), non-uniformity of cracks, surface texture (e.g., tining), lack of sufficient background illumination, and presence of other features such as joints, among others, continue to keep this area of research active with researchers constantly seeking newer methods and algorithms to address these challenges. Not surprisingly, the recent achievements by so-called deep learning (DL) algorithms have caught the attention of the pavement image analysis community and have inspired them to move from systems that use handcrafted features to data-driven distress detection systems (i.e., systems that automatically learn features from the images). Because of the availability of inexpensive, parallel hardware, and massive amounts of unlabeled data, deep learning has already produced breakthrough results in computer vision, speech recognition, and text processing. A popular example of DL success is its deployment in self-driving cars.

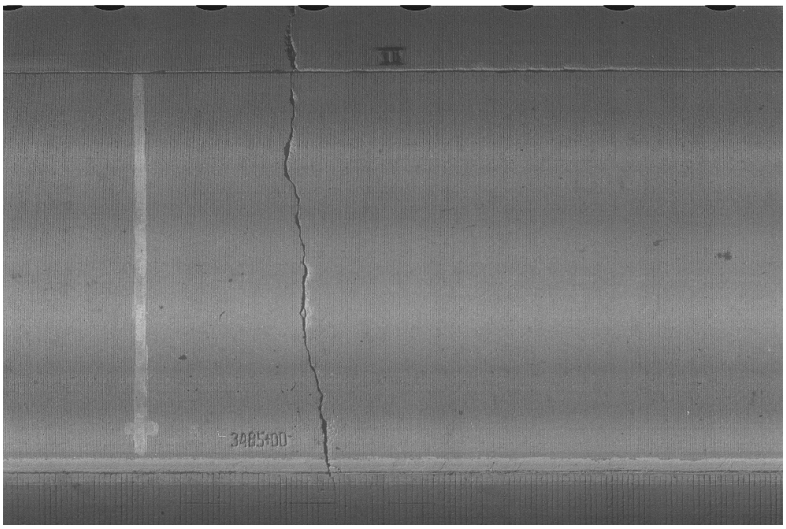

Figure 1. A sample Portland cement concrete (PCC)-surfaced pavement distress image captured using pavement data collection vehicle moving at highway speed and equipped with a downward-looking high-speed digital camera (Source: FHWA LTPP database).

To date, applications of DL to pavement image analysis have mainly employed convolutional neural networks (CNNs or ConvNets), a specific DL model with many convolution layers. Deep CNNs (DCNNs) are characterized by deeper architectures with numerous hidden layers enabling them to learn many levels of abstraction, as opposed to shallow architectures with typically fewer hidden layers [6–8]. Although the first published works on the application of CNNs or DL in general to pavement crack detection appeared in 2016, some 12 papers have already been published between 2016 and beginning of 2018. Although not as steep a growth in research productivity as in other areas such as medical image analysis, it is evident that the interest in the application of DL to address various challenges in vision-based automated pavement distress detection is fast growing.

This paper provides the first narrative review on the application of deep learning to pavement image analysis and automated distress detection. Although there is not yet a sufficient number of papers on this topic to conduct a comprehensive survey in the traditional sense, this quick review nonetheless covers enough ground to assess the state-of-the-art and will hopefully spur future research in the application of DL to pavement image analysis. Considering the rather narrow range of studies published so far on this topic, peer-reviewed journal articles, as well as articles appearing in conference proceedings, were included in this narrative review. Additionally, one preprint appearing in arXiv online repository was also reviewed.

The rest of this review paper is structured as follows. In Section 2, we summarize the existing and emerging deep learning software frameworks for computer vision applications, especially relevant to

automated pavement distress detection. Section 3 forms the main crux of this paper, where each study reviewed in this paper is briefly summarized. Section 4 provides useful overview and comparison tables of all publications in terms of objectives and datasets, network architecture and hyper-parameters, software framework used, hardware specs, and summary of test results. We conclude with a summary and some emerging areas of future research driven by ongoing advances in deep learning technology.

2. Some Existing and Emerging Deep Learning Frameworks for Computer Vision Applications

A significant amount of parallelism in computations is involved in DCNN implementations. To facilitate this, a number of popular open-source DL software frameworks exist, such as Caffe, the Microsoft Cognitive Toolkit (CNTK), Google's TensorFlow, Theano, Torch, dmlc MXnet, Chainer, and Keras, among other. These frameworks make use of the system's hardware (CPU and/or GPU) settings to implement parallel programming and accelerate the computational process. A brief overview of the most commonly used DL software frameworks for computer vision applications (especially relevant to crack detection) is presented in this section, highlighting their pros and cons. This is mostly based on some recently published comparative studies on DL software frameworks by evaluating these frameworks in terms of extensibility, hardware utilization, and speed [9–11]. It is worth noting that all these DL frameworks are undergoing constant development with active contributions from researchers and the open-source community, and therefore the study results and conclusions from the reported comparative studies may not be up to date.

2.1. Caffe

Caffe is developed at the Berkeley AI Research (BAIR) center and the Berkeley Vision and Learning Center (BVLC) at the University of California, Berkeley with "expression, speed, and modularity in mind" [12]. It is considered to be an easy-to-deploy production platform developed exclusively for DL-based computer vision systems and is believed to be one of the fastest ConvNet or CNN implementations available with an ability to process over 60 million images per day [13].

2.2. TensorFlow

TensorFlow, originally developed by researchers and engineers working on the Google Brain Team, uses data flow graphs for numerical computation and is mainly designed for developing and implementing deep neural network models [14]. One major advantage of TensorFlow that vastly increased its popularity among DL researchers and companies is its ability to deploy computation to one or more CPUs/GPUs on a variety of systems and devices through a single application programming interface (API) [10]. Based on a comparative study of Theano, Torch, Neon, and TensorFlow DL frameworks, Bahrampour et al. [9] concluded that TensorFlow, although a very flexible framework, is not as competitive as other studied frameworks in terms of its performance on a single GPU. However, in a similar benchmarking study comparing Caffe, CNTK, TensorFlow, and Torch, Shi et al. [11] reported that no single framework consistently outperforms others. Fonnegra et al. [10] reported that TensorFlow was faster than Theno when the DL architecture contained long short term memory (LSTM) units as its core.

2.3. Theano

Theano, named after the Greek mathematician who may have been Pythagoras' wife, was originally developed at the Montreal Institute for Learning Algorithms (MILA) as a "CPU and GPU compiler in Python" to "support rapid development of efficient machine learning algorithms" [15]. Although not intended to be a specific DL framework, it is a Python library for speedy numerical computations and is considered to be a forerunner of today's open-source DL frameworks, such as CNTK and TensorFlow. The efficient symbolic differentiation (ability to compute derivatives for functions with one or more inputs) offered by Theano is considered to be a big advantage for implementing non-standard DL architectures [9]. For GPU-based training and deployment of both

CNNs and LSTM recurrent neural networks (RNNs), Bahrampour et al. [9] reported that Theano resulted in the best performance, in comparison with Torch and TensorFlow, for smaller networks. As of October 2017, after almost ten years of development, it was reported that there will be no further development of Theano after Version 1.0.

2.4. Torch

Torch, based on LUA programming language, is a scientific computing platform for building machine learning algorithms with fast and efficient GPU support [16]. Since its initial release in 2002, Torch has grown to be a popular open-source DL framework for commercial applications (Facebook, IBM, etc.), academic research studies, and so on, owing to its simplicity and extensibility [10]. Based on the results of a comparative study by Bahrampour et al. [9], Torch performed the best for CPU-based training and deployment of any DL architecture that was tested.

2.5. Keras

Keras is a high-level Python DL library and API capable of running on top of TensorFlow, CNTK, or Theano as the backend [17]. It is well known, among both budding DL researchers and experienced ones, for its ease-of-use (minimal programming) and ability to allow fast prototyping. Like other open-source DL software frameworks, Keras is built on the guiding principles of user-friendliness, modularity, and extensibility.

3. Application of Deep Learning to Vision-Based Pavement Distress Detection

Computer vision-based automated pavement distress detection has been a subject of high interest to the pavement engineering and computer vision community over the past decades. The development of vision-based automated pavement crack detection methods over the years is well-documented in the literature: intensity-thresholding [18–21], edge detection [22], wavelet transforms [23–25], texture-analysis, and machine learning [26–29] techniques [30].

In recent years, there has been a steady increase in interest among researchers and companies to employ DCNN or DL for automated pavement distress detection. The following section will highlight some recently completed or ongoing research efforts on the application of DL or its variant to automated pavement crack detection.

3.1. Detecting No-Crack Surfaces from Mobile Mapping Images

Street view image databases generally contain disproportionately larger number of images with features of non-interest (such as cars, sidewalks, surfaces with no distresses, etc.) compared with those with features of interest. Some [31] employed DL to detect and eliminate such images (especially road surfaces with no cracks) from a database of 9712 images acquired using a mobile mapping system, to reduce the street view image database size for further processing. The NVIDIA Deep Learning GPU Training System (DIGITS) version 2.0 software, an open-source web-based interactive DL GPU training system, was used to perform all DL-related operations. DIGITS integrates the Caffe DL framework [13] from the Berkeley Learning and Vision Center, and it supports GPU acceleration using cuDNN, a GPU-accelerated library of primitives for DCNNs. A batch size of 13 was used and the number of epochs was set to 50 during the DL experiments.

For the sake of comparison, crack detection was also carried out using the CrackIT software, a Matlab toolbox for road crack detection and characterization based on image processing and pattern recognition techniques [32]. It was reported that the DIGITS DL framework yielded 3% higher precision compared with the CrackIT framework, while it achieved 20% higher recall compared with the CrackIT framework. When it came to street view images with problematic features such as oil spills, horizontal and vertical shadows, and unevenly scattered shadows, both methods failed to give accurate results. Some [31] also analyzed the total time spent in developing each of the algorithms while noting the differences in time distribution phases between image processing and deep learning methods. CrackIT

takes similar computing time for each image (around half a minute). On the other hand, much human time goes in annotating the images for classification using DIGITS DL. However, once the DL model was developed, it took only 2.5 ms to analyze one image. It was shown that DL is especially more time efficient when the number of images exceeded 8000 [31].

Some [31] concluded that DL has significant potential in automated pavement crack detection and classification especially when it comes to "big data" image analysis. Some [31] also suggested to carry out image acquisition on a cloudy weather or the addition of an illumination device to the mobile mapping system to avoid shadows and increase the image quality. Because the accuracy of DL based classification is dependent on the quantity of images used in training, Some [31] suggested using at least 10,000 images in each class for better classification accuracy. Extending the DL-based crack identification analysis to one of object detection analysis, especially with street view images and crack severity classification (low, medium, and high), was also recommended as a potential area for future investigation.

3.2. Crack Detection from Low-Cost Smartphone Pavement Images

In what appears to be one of the earliest works on the direct application of DCNN to road crack detection, Zhang et al. [33] employed DCNN or ConvNet for classifying square image patches with or without cracks based on 500 pavement images of size 3264 × 2448 collected using a low-cost smartphone. The authors used a sampling strategy to generate one million three-channel (RGB) 99 × 99 pixel image patches (640,000 samples for training, 160,000 samples for five-fold cross-validation during the training, and 200,000 for testing) from the 500 collected pavement images. The dropout regularization method was used between two fully connected layers of the ConvNet, constructed via the Caffe DL framework [13], to reduce over-fitting during training. The rectified linear units (ReLU) activation function was used with the ConvNet, as is typically done with DCNNs to further speed-up the training process. The experiments were performed on an Intel® Xeon® with 8 GB RAM and NVidia Quadro K220 GPU. The stochastic gradient descent (SGD) optimization method was used during training with a batch size of 48, momentum of 0.9, and weight decay of 0.0005. The minimum error was achieved during training in less than 20 epochs [33].

The trained CrackNet provided a crack probability map when applied on a new test image, and the final crack probability of the test patch image was re-estimated using a threshold that maximized the F_1 score on the validation images. For the sake of comparison, traditional machine learning classifiers such as support vector machine (SVM) and boosting methods were also applied on the vectorized patch images with hand-crafted features such as the mean RGB and saturation histogram, among others. Zhang et al. [33] demonstrated that their developed ConvNet, which has automatically learned to discriminate between image patches with and without cracks and without consideration of the pavement geometry, was able to outperform SVM and the boosting methods in crack detection of patch images with a precision of 0.8696, recall of 0.9251, F_1 score of 0.8965, and an area-under-the-ROC-curve (AUC) of 0.9592 [6].

3.3. Effect of DCNN Depth on Pavement Crack Detection Accuracy

Pauly et al. [34] hypothesized that shallow DCNNs, such as the four-layer DCNNs used by Zhang et al. [33], for vision-based pavement crack detection cannot fully harness the power of DCNNs, and thus set out to demonstrate the effectiveness of deeper networks in improving the accuracy of crack detection. Their study employed a generic DCNN architecture with four convolutional layers, four max pooling layers, and two fully-connected layers. The SGD optimization algorithm was used to minimize the categorical cross entropy loss function. A batch size of 48, learning rate of 0.0001, decay of 0.0005, and momentum of 0.9 was used. Two training experiments were conducted to meet the objectives of the study and the training was completed at 80 and 40 epochs, respectively. The dataset consisted of the same 500 pavement images collected using a low-cost smartphone by Zhang et al. [33]. While Experiment 1 focused on studying the impact of network depth on crack detection accuracy, Experiment

2 focused on examining the variation in location between the training and testing set (i.e., train on images from one location and test on images from another location) on the crack detection performance. While a computer node at the high performance computing (HPC) facility at the University of Leeds was used for network training, the testing was carried out on an Intel® Xeon® desktop workstation with 128 GB RAM and Nvidia Quodro M4000 GPU. The DCNN framework was implemented in Keras with TensorFlow as the backend. Based on their findings, Pauly et al. [34] concluded that increasing the depth of DCNNs does lead to better crack detection performance in terms of accuracy and recall, although they could not define a threshold. Further, the system failed to perform well when the location of the pavement images used in training and testing were different [34].

3.4. Generalization of DCNN on Large Open-Source Pavement Distress Dataset

Eisenbach et al. [35] introduced the German asphalt pavement distress (GAPs) dataset as the first free and publicly available massive pavement distress images dataset suitable for training high-performance DCNNs. Almost all previous studies in this domain used their own datasets collected and annotated differently, and thus the performance of the developed distress detection algorithms and techniques could not be compared on a standard benchmarking dataset. Even when the pavement distress datasets were made publicly available, they were not big enough for direct implementation of the DCNNs [35]. Thus, the GAPs dataset (available at http://www.tu-ilmenau.de/neurob/data-sets-code/gaps/) seems to be the first attempt at creating a standard benchmarking pavement distress images dataset for deep learning applications. It includes 1969 grayscale pavement images (1418 for training, 51 for validation, and 500 for testing) with various distresses, including cracks (longitudinal/transverse, alligator, sealed/filled), potholes, patches, open joints, and bleeding [35].

For their deep learning experiments, Eisenbach et al. [35] first employed the same four-block ConvNet as that used by Zhang et al. [33], and refer to it as RCD net (convolutional neural network for road crack detection). Because one of main objectives of their study was to study the effect of regularization techniques on the generalization ability of the DCNN, and because RCD net does not need much regularization, Eisenbach et al. [35] opted for a more deeper network, referred to as ASINVOS (eight convolutional layers, three max-pooling layers, and three fully-connected layers) with 4.0 Million weights, making it an ideal case for regularization experiments. The effect of both dropout and batch normalization regularization techniques on network performance was evaluated separately and in combination. Two additional regularization techniques, namely the max–norm regularization and weight decay were also evaluated. It took approximately three months on two NVIDIA Titan X GPUs to complete all the experiments on Keras DL framework with Theano as the backend. Based on the findings, Eisenbach et al. [35] reported that the proposed DL approaches outperformed state-of-the-art vision-based distress detection approaches. The generalization ability of the network was better when using either dropout or batch normalization [35].

3.5. RoadDamageDetector: A DL Mobile App for Road Damage Detection Based on Open-Source Smartphone Road Images

Maeda et al. [36] employed an end-to-end DL-based state-of-the-art object detection method for detecting and distinguishing objects in road images, acquired using a smartphone installed in a moving car under realistic weather and lighting conditions, into eight different output categories (five kinds of cracks, rutting-bump-pothole-separation, white line blur, and cross walk blur). Furthermore, Maeda et al. [36] have also made publicly available the entire large-scale dataset consisting of 9053 road distress images (with 15,435 instances of road distresses), as well as the DL-based smartphone application, RoadDamageDetector, at this website: https://github.com/sekilab/RoadDamageDetector/.

A strong motivation for the development of the RoadDamageDetector [36] is the strong need felt by municipalities in Japan (also true for local highway agencies across the globe) to not only detect the existence of distress from a road image (which its predecessor, the Lightweight Road Manager [37], did), but also detect and distinguish different types of damage (cracks, rutting, etc.)

using low-cost resources (such as a smartphone app) so that appropriate follow-up maintenance activities can be pursued to address the distressed pavement [36]. For this study, Maeda et al. [36] developed a smartphone app that captures road images of 600 × 600 pixels once per second when the smartphone is mounted on the dashboard of a car driving at an average speed of 40 km/h (25 mph). As the ultimate goal was a DL-based smartphone app for road damage detection, the single shot multibox detector (SSD) [38] with Inception V2 and SSD using MobileNet [39] DL frameworks were chosen for this study as they are considered efficient network architectures (very low computational burden) and implementations for mobile vision applications [36]. For training SSD Inception V2, they had to downsize the images from 600 × 600 pixels to 300 × 300 pixels and used an initial learning rate of 0.002, reduced at a 0.95 rate of decay every 10,000 epochs. For training the SSD MobileNet, the input image size again was 300 × 300 pixels and they used an initial learning rate of 0.003 with a similar rate of decay as SSD Inception V2. Experiments were performed on Ubuntu 16.04 OS with 15 GB RAM and an NVIDIA GRID K520 GPU. For the development of the smartphone app, a Nexus 5X smartphone with an MSM8992 CPU and 2 GB RAM was used. The best-performance results revealed that the RoadDamageDetector smartphone app was able to achieve recalls and precisions greater than 75% with an inference time of 1.5 s [36].

3.6. Measurement and 3-D Reconstruction of Concealed Cracks in Asphalt Pavements Using GPR Images

When a new asphalt concrete (AC) layer is overlaid over existing (distressed) Portland cement concrete (PCC) pavement or a base layer (semi-rigid base), it often leads to a phenomenon referred to as reflection cracking, where a hidden crack initiating in the bottom PCC base/pavement layer propagates to the top AC layer as a result of repeated traffic and environmental loading [40]. Thus, proactive preventive measures for reflection cracking involve detection and location of concealed cracks in the bottom PCC base/pavement when they have not yet propagated to the top. Ground penetrating radar (GPR), a promising technology for non-destructive evaluation of pavement structures with several useful subsurface applications, has been used for characterizing subsurface pavement defects like concealed cracks. However, the state-of-the-art defect detection methods for GPR images involve complex manual processes, and thus there is a need for GPR-based automated, low-cost damage characterization method [41]. To fulfill this need, Tong et al. [41] employed separate CNN models for the automatic recognition, location, measurement, and 3D reconstruction of concealed cracks from GPR images of asphalt pavement.

A total of 6832 GPR images with various damage types as class labels ('concealed cracks', 'subgrade settlement', 'roadbed cavities', and 'no damage') were used for training and testing the CNNs implemented using the Caffe DL environment in a PC with Intel® Core™ i7-6700 CPU, 8 GB RAM, and NVIDIA GeForce GTX GPU [41]. The damage recognition CNN had a typical CNN architecture used for image classification problems: input layer that accepts input images of size 256 × 256 pixels, two convolutional layers, two subsampling layers (max-pooling), two fully-connected layers, and one output layer. The location CNN had a similar structure as the recognition CNN, except that the input images were scaled to a size of 64 × 64 pixels. The feature extraction CNN (whose outputs were used for 3D reconstruction) had a similar architecture as the location CNN, except that seven output neurons were used to represent seven shape coordinates for the damage features. Based on the study results, Tong et al. [41] concluded that all three CNNs developed for recognition, location, and feature extraction of concealed cracks achieved their purposes and are suitable for field pavement applications.

3.7. Segmented Grid Based Pavement Crack Classification with DL and PCA

Wang and Hu [42] proposed a CNN-based pavement crack classification method by first segmenting the pavement image into grids of different sizes, detecting the presence of cracks, and then classifying the crack type (longitudinal, transverse, and alligator crack) after analyzing the distribution of grids using principal component analysis (PCA). For their study, RGB pavement images (3264 × 2448 pixels) were captured with an iPhone 6 smartphone by maintaining the distance

between the smartphone camera and the pavement to be approximately 1.3 m (4.3 ft). The RGB images were then downsized to grayscale images of 960 × 704 pixels, suitable for analysis. Two grid settings were applied on each pre-processed image: 15 × 11 non-overlapping grids, with each grid holding 64 × 64 pixels; and 30 × 22 grids, with a grid size of 32 × 32 pixels.

The CNN architecture implemented by Wang and Hu [42] in the TensorFlow DL framework consisted of two convolutional layers, two sub-sampling layers (max-pooling), and one fully-connected output layer. They employed the tanh activation function as opposed to the famous ReLu activation function used by most vision-based image classification studies. The learning rate was set to 0.1 and the batch size to 32. By applying the trained CNN for crack detection on segmented pavement images, they derived the skeleton of cracks by retaining only the grids containing crack pixels and calculating the coordinates of crack regions. PCA is then applied on the distribution of grids containing cracks to estimate the specific crack type: longitudinal, transverse, and alligator. Based on their study results, Wang and Hu [42] reported that segmentation of pavement images with a grid size of 64 × 64 achieved better classification performance with the following correct rate of crack classification: 97.2% (longitudinal crack), 97.6% (transverse crack), and 90.1% (alligator crack).

3.8. Learning the Structure of Pavement Cracks from Raw Image Patches

Fan et al. [43] proposed an automated pavement crack detection methodology based on structured prediction using CNN modeled as a multi-label classification problem. Their proposed methodology has the ability to predict the whole crack structure at the pixel level in a pavement image based on learning the crack structure of a very small patch within the image. First, a training database is built by extracting individual pixel-level patches from raw images to train a CNN. As the ratio of crack-pixels to non-crack-pixels is quite low for typical crack images, Fan et al. [43] also introduced a novel strategy to address the challenge of severely imbalanced samples. By summing the centered-patch-structure predictions of the trained CNN applied on all pixels, a probability map is obtained that reflects the overall pavement condition.

The dataset for their experiments came from two established pavement images databases: CFD [44] and AigleRN [18]. Fan et al. [43] employed a typical CNN architecture with four convolutional layers, two sub-sampling (max-pooling) layers, and three fully-connected layers. All hidden layers were equipped with ReLu units and the output layer with sigmoid activation function. The CNN was implemented in a TensorFlow DL environment and the experiments were carried out on an Intel Xeon E5-2690 workstation with 2.9 GHz CPU, 64 GB RAM, and NVIDIA Quadro K5000 GPU. A dropout ratio of 0.5 was used and the weight decay rate was set to 0.0005 during the experiments. The Adam optimizer was employed with a default learning rate of 0.001 and the batch size was set to 256. Based on their study results, Fan et al. [43] reported that their proposed methodology achieved better crack detection performance compared with other state-of-the-art methods, especially in dealing with different pavement textures, and has a better generalization ability (training on one database and testing on another).

3.9. Continuous Pavement Inspection with CNN Trained on Google Street View Images

Ma et al. [45] propose an innovative method for large-scale image-based pavement degradation assessment using CNN, Fisher vector encoding, and UnderBagging random forests. While most studies reviewed so far on this topic focused on pavement images acquired with data collection vehicles or smartphones, the study by Ma et al. [45] combined the publicly available maintenance records of road infrastructure (along with GPS coordinates) with GPS-localized Google Street View images to create a readily-labeled, large-scale road images dataset. To overcome the challenges associated with this atypical dataset for pavement degradation assessment (such as texture classification when there is large variation, class imbalance, etc.), Fisher vectors with CNN (FV–CNN) and UnderBagging random forests were employed to develop an automated pavement condition rating method.

For their study, Ma et al. [45] collected more than 700,000 images of road surfaces from about 70,000 street segments recorded in the New York City Department of Transportation. Using their

proposed methodology, they were able to achieve an accuracy of 59.2% on the proposed complex dataset. Their dataset and the code are publicly available for download at the following site: http://www3.cs.stonybrook.edu/~cvl/pavement.html.

3.10. Pixel-Level Crack Detection on 3D Asphalt Pavement Surfaces

Zhang et al. [46] proposed CrackNet, an efficient DCNN for pixel-perfect crack detection on 3D asphalt pavement surfaces. After highlighting the limited success of state-of-the-art crack detection algorithms for 2D pavement images, Zhang et al. [46] discussed the emergence of 3D laser imaging technology for automated pavement distress data collection and assessment and the shortcomings of recent 2D and 3D image-based distress detection algorithms in achieving consistent and satisfactory levels of precision and bias with varying pavement conditions.

Based on a brief review of some recent research studies on the use of DCNN for vision-based crack detection, Zhang et al. [46] concluded that the DCNNs or the DL approach used in these studies, despite their success in demonstrating the potential of DL for automated crack detection, could not achieve pixel-perfect accuracy, mainly because the pooling layers in the CNN tend to compress the original pixel-level data with increasing levels of abstraction, leading to information loss. Thus, a key difference between their DCNN and those used in previous studies is that their proposed network does not have any pooling layers (although min-pooling is used at the data preparation stage in downsizing the original 3D images to reduce computational overhead) and it enables pixel-level training of the classifier. The data consisted of 1800 observations of 1-mm 3D asphalt surface images (representing various textures and mix types, including hot-mix asphalt [HMA] and warm-mix asphalt [WMA]) from the PaveVision3D system for training, and another 200 for testing the system. Their proposed CrackNet architecture consisted of two fully connected layers, two convolutional layers (with one acting as a 1×1 convolutional filter), and one output layer. The mini-batch gradient descent optimization was used and the mini-batch ranged from 10 to 30, while the learning rate ranged from 0.001 to 0.05. The dropout regularization technique was used with a cut-off value of 0.5. CrackNet was programmed in C++ using a parallel computing platform and two GPU devices. Based on the results and findings, it was concluded that CrackNet significantly outperforms state-of-the-art image processing and machine learning based classifiers, and needs to be further improved for detecting hairline cracks [46].

3.11. Automated Crack Detection with Pre-Trained DL Model Using Transfer Learning

Gopalakrishnan et al. [47] proposed the use of a pre-trained DCNN model with transfer learning for automated pavement crack detection. Large annotated image datasets are typically required by DCNNs to achieve accurate and generalized predictions. It is not always possible to have access to large pavement images datasets, although some very recent studies reviewed in this section have made them available publicly to spur innovative research in this area. In many domains, acquisition of data and annotating them, especially at large scale, is cost-prohibitive. The use of 'off-the-shelf' DCNN features of well-established DCNNs such as VGG-16 (16-layer DCNN developed by the Visual Geometry Group (VGG) at the University of Oxford), AlexNet, and GoogLeNet pre-trained on large-scale annotated natural image datasets (such as ImageNet) has worked well for most cross-domain image classification problems through the concept of transfer learning and fine-tuning [48]. Transfer learning enables the deployment of deep learning models trained on "big data" natural-image datasets (like ImageNet) and "transfers" their learning ability to a new domain, rather than freshly train a DCNN classifier from the beginning [49]. This highly repurposable nature of deep learning models has been amply demonstrated through several cross-domain image classification studies, especially in the area of medical image analysis [50].

An interesting aspect of the research study by Gopalakrishnan et al. [47] was that the truncated VGG-16 DCNN, pre-trained on massive ImageNet database that contains millions of natural images, was used to vectorize the labeled pavement images, and a machine learning classifier was then used to predict the labels ('crack' or 'no crack'). In this strategy [17], (1) only the convolutional part of the

VGG-16 DCNN (everything up to the fully-connected layers) is instantiated; (2) the model is run once with the training and validation data, and the last activation maps before the fully-connected layers or the "bottleneck features" from the VGG-16 DCNN are recorded; and (43) a small fully-connected classifier is finally trained on the stored features. The dataset (about 1056 pavement images) used in their study came from the Federal Highway Administration's (FHWA's) Long-Term Pavement Performance (LTPP) database and it had a mixture of AC-surfaced and PCC-surfaced pavement images. They used the Keras DL framework and the experiments were carried out in a PC equipped with Intel® Core™ i7-5600 CPU @ 2.60 GHz (20 GB RAM) with no GPUs on 64-bit Windows® 10 OS. The original input images with a size of 3072 × 2048 pixels were downscaled to 1000 × 500 pixels to reduce the computational burden. For the single-layer NN classifier implemented in Keras, 256 neurons were used in the hidden layer. The dropout value was set to 0.5. The image batch size was set to 32 and all models were trained for up to 50 epochs. A single-layer NN classifier (with 'Adam' optimizer) trained on ImageNet pre-trained VGG-16 DCNN features yielded the best performance [47].

3.12. Sealed Crack Detection with Transfer Learning and Fine-Tuning

In AC-surfaced and PCC-surfaced pavements, cracks are filled and sealed as part of routine maintenance activity to prevent ingress of water and incompressible material into the cracks. During an automated visual pavement distress survey and post-processing, there is often a need to distinguish between sealed cracks and unsealed cracks for classifying crack severity and consequently budgeting—an issue that has mostly been neglected in studies dealing with the development of automated crack detection algorithms [51]. To address this important issue, as well as the challenges caused by varying pavement textures and intensity inhomogeneity, Zhang et al. [51] proposed a unified CNN-based framework to detect and separate cracks from sealed cracks with the power of transfer learning.

For their study, Zhang et al. [51] used a total of 800 (2000 × 4000 pixels) low-similarity pavement images, among which 500 were used to generate the training block pixels (30,000 crack blocks, 30,000 sealed crack blocks, and 30,000 background blocks) and the remaining 300 images were used to generate 20,000 crack blocks, 20,000 sealed crack blocks, and 20,000 background blocks through image resizing and rotation data augmentation techniques. The experiments were carried out on a HP workstation with 8 GB RAM and NVIDIA Quadro K4000 GPU. The two-step preclassification procedure proposed by Zhang et al. [51] and implemented in Caffe [13] DL framework first transfers the generic knowledge (from "natural images dataset" domain to "pavement images dataset" domain) from the first convolutional layer of the ImageNet pre-trained DCNN model by keeping them unchanged during training, and fine tunes the parameters of the other convolutional layers using the target pavement images. After the pavement images are preclassified into cracks, sealed cracks, and background regions, a blockwise thresholding method and tensor voting-based curve detection are used to extract the cracks and/or sealed cracks. Zhang et al. [51] reported that their proposed method gave better performance than traditional edge detection and other state-of-the-art approaches such as CrackIT [32] and CrackForest [44].

3.13. Other Related Studies

Cha et al. [52] proposed a CNN-based method for detecting cracks on concrete images captured under uncontrolled conditions using a hand-held camera. It was reported that their proposed method was especially good at detecting thin cracks under lighting conditions that are hard to detect using traditional methods like Canny and Sobel edge detection. Feng et al. [53] proposed a deep active learning strategy for civil infrastructure defect detection and classification where a deep residual network (ResNet) was used to train a small set of images with defect labels and use this low-accuracy defect detector to filter out many non-defect images. Gopalakrishnan et al. [54] proposed the use of pre-trained deep learning models with transfer learning for crack damage detection in UAV images of civil infrastructure (which also included a small proportion of road surface images). The results show

that their proposed method can rapidly and easily achieve up to 90% accuracy in finding cracks in realistic situations without any data augmentation and preprocessing.

4. Discussion

Deep learning is fast becoming a successful alternative approach for vision-based pavement crack detection. A total of 12 recently published papers were reviewed in this study, among which two were published in 2016, seven in 2017, and two in 2018 (as of March). Based on the current interest and continued success of DL technology, it can be easily expected that by the end of 2018, these numbers will be doubled, if not tripled.

4.1. Objectives and Datasets

An overview of the major objective and the dataset used in each of the reviewed studies is presented in Table 1. Almost all the studies focused on 2D images were acquired either using smartphone or a dedicated data collection vehicle equipped with a camera. Thus far, the study by Zhang et al. [46] is the only study that focused on 3D asphalt pavement surface images. Interestingly, the study by Tong et al. [41] focused on GPR images to detect concealed cracks, whereas the other studies focused on detecting pavement surface distresses. Among the 2D images dataset category, seven are private, while five of them are publicly available. Among the ones available for free download, at least two of them consist of street view images (i.e., images that contain other objects in addition to the road surface). This is an interesting trend as vision-based distress detection is entering an era of open-source deep learning. While most previous studies on this topic tended to focus on authors' own datasets that are not publicly available for benchmarking the performance of developed algorithms, it is encouraging to see large-scale distress datasets made publicly available for training well-performing deep neural networks and for doing comparison studies.

Table 1. Overview of objectives and datasets used by papers. DL—deep learning; GPR—ground penetrating radar; FHWA/LTPP—Federal Highway Administration's Long-Term Pavement Performance database.

Reference	Dataset	Goal
Some [31]	Street view images (private)	Decrease dataset by detecting no-crack road surfaces (image-level)
Zhang et al. [33]	Smartphone images (private)	Crack detection from smartphone images (image-level)
Pauly et al. [34]	Smartphone images [33]	Use of deeper convolutional neural network (CNNs) for pavement crack detection (image-level)
Eisenbach et al. [35]	German asphalt pavement distress (GAPs) (public)	Generalization of CNN on large open-source pavement distress data (block-level)
Maeda et al. [36]	Smartphone street view images (public)	A DL mobile app for road damage detection (image-level)
Tong et al. [41]	GPR images (private)	Location and measurement of concealed cracks in asphalt pavements (block-level)
Wang and Hu [42]	Smartphone images (private)	Segmented grid based pavement crack classification (block-level)
Fan et al. [43]	CFD (Shi et al., 2016) [11]; AigleRN (Chambon and Moliard, 2011) [18] (public)	Learning the structure of pavement cracks from raw image patches (pixel-level)
Ma et al. [45]	Google StreetView images (public)	Continuous pavement inspection "in the wild" (image-level)
Zhang et al. [21]	3D asphalt surface images from PaveVision3D system (private)	Pixel-level crack detection on 3D asphalt pavement surfaces (pixel-level)
Gopalakrishnan et al. [47]	FHWA/LTPP (public)	Crack detection with pre-trained DL model using transfer learning (image-level)
Zhang et al. [51]	Local (private)	Sealed crack detection with transfer learning and fine tuning (block-level)

4.2. Network Architecture and Hyper-Parameters

An overview of the DL network architecture and the associated training parameters used in each of the papers is summarized in Table 2. It is not surprising that most of the reviewed papers used a typical CNN architecture with some variations, either inspired by LeNet or VGG-net, as these networks have already demonstrated their success in several image classification problems. One slight variation to the CNN architecture was introduced by Zhang et al. [46], where no sub-sampling or max-pooling layers were used in the network architecture. As their goal was pixel-level crack detection, they wanted to retain as much image information as possible during the training process, which would otherwise occur by the use of max-pooling layers. The use of SSD Inception V2 and SSD MobileNet by Maeda et al. [36] is also a fairly new concept for computationally-efficient distress detection from pavement images. The hybrid Fisher vector (FV)–CNN architecture was used by Ma et al. [45] mainly to achieve better representation of road surfaces in their street view images dataset, which also included other objects such as poles and cars, among others.

There are many CNN hyper-parameters that need to be tuned for achieving best performance. Most papers reviewed in this study appeared to use the stochastic gradient descent (SGD) or mini-batch gradient descent (MBGD) optimization. The use of the dropout technique as a regularization mechanism to avoid over-fitting has become a standard practice in modern CNN designs and not surprisingly, most of the reviewed papers used it. Similarly, except for one or two studies, which used the sigmoid or tanh activation function in their hidden layers, all others used ReLU, the now de facto standard activation function of deep learning. Mini-batch size is often referred to as batch size for brevity and it is usually limited by the CPU or GPU memory requirements. Among the papers reviewed in this study, batch size ranged from 10 to 400.

Table 2. Overview of deep learning architecture and hyper-parameters used by individual studies.

Reference	Network Architecture	Hyper-Parameters
Some [31]	CNN: N/A	B = 13; E = 50
Zhang et al. [33]	ConvNet (inspired by LeNet): 4 C; 4 MP; 2 FC; 1 O	Op = SGD; R = Dropout (0.5); A = ReLU; B = 48; E = 20; M = 0.9; WD = 0.0005
Pauly et al. [34]	ConvNet (inspired by LeNet): 5 C; 4 MP; 2 FC; 1 O	Op = SGD; R = Dropout; A = ReLU; B = 48; E = 40–80; M = 0.9; LR = 0.0001; WD = 0.0005
Eisenbach et al. [35]	ASINVOS (inspired by VGG-net and AlexNet): 8 C; 3 MP; 3 FC	Op = SGD; R = Dropout (0.1–0.5); A = ReLU; B = 256; M = 0.7; LR = 0.01
Maeda et al. [36]	SSD Inception V2 SSD MobileNet	SSD Inception V2: LR = 0.002 (LR decay = 0.000095) SSD MobileNet: LR = 0.003 (LR decay = 0.000095)
Tong et al. [41]	Recognition CNN: 2 C; 2 MP; 2 FC; 1 O	Op = Back-propagation; A = Sigmoid; B = 100; E = 30
Wang and Hu [42]	CNN: 2 C; 2 MP; 1 FC	A = tanh; B = 32; LR = 0.1
Fan et al. [43]	CNN: 4 C; 2 MP; 3 FC	Op = Adam; R = Dropout (0.5); A = ReLU; B = 256; LR = 0.001; WD = 0.0005; E = 13–43
Ma et al. [45]	Fisher Vectors with CNN (FV–CNN) using VGG-D (Simonyan and Zisserman, 2014)	N/A
Zhang et al. [21]	CNN: 2 C; 2 FC; 1 O	Op = MBGD; R = Dropout (0.5); B = 10–30; LR = 0.001–0.05; E = 700
Gopalakrishnan et al. [47]	Truncated VGG-16 + transfer learning classifier	Classifier: Op = Adam; R = Dropout (0.5); A = ReLU; B = 32; E = 50
Zhang et al. [51]	CUDA-ConvNet (Krizhevsky et al. 2012) + transfer learning: 5 C; 3 FC; 1 O	Op = MBGD; LR = 0.001; WD = 0.005; B = 400; E = 40,000

Note: CNN or ConvNet = convolutional neural network; C = convolutional layer; MP = max-pooling layer; FC = fully-connected layer; O = output layer; Op = optimizer; SGD = stochastic gradient descent; MBGD = mini-batch gradient descent; R = regularization; A = activation function; ReLU = rectified linear units; B = batch size; E = epochs; M = momentum; LR = learning rate; WD = weight decay; N/A = not available or not applicable; SSD = single shot multibox detector.

4.3. Software Framework, Hardware Specs, and Test Results Summary

An overview of the DL software framework, hardware specs, and summary test results of the reviewed papers are presented in Table 3. Based on the limited number of published papers, thus far, all three DL software frameworks, viz., Caffe, TensorFlow, and Keras, are more or less equally popular for pavement distress image classification (Figure 2).

Most studies employed the use of NVIDIA GPUs to accelerate the computations during deep learning training as training CNNs involve lots of matrix multiplications and vector operations that can be parallelized in GPUs. Gopalakrishnan et al. [47] did not use any GPUs as a pre-trained DCNN was used as a feature extractor and the feature vectors (i.e., ImageNet pre-trained VGG-16 bottleneck features) were then used to directly train a new classifier using the transfer learning concept. Thus, the weights of the bottom layers of the VGG-16 DCNN were frozen while vectorizing the pavement images. On the other hand, Zhang et al. [51], who also employed transfer learning, required the use of the GPUs because they only transferred the learning from the bottom-most convolutional block (i.e., most abstract representation) of the ImageNet pre-trained DL model and fine-tuned the weights of the other convolutional blocks for the target domain.

Similar to GPUs, Google has been working, since 2016, on tensor processing unit (TPU) as a proprietary, non-commercial artificial intelligence (AI) accelerator application-specific integrated specific circuit (ASIC), a chip specifically designed to work with and accelerate TensorFlow computations. It is also worth mentioning that there are now a number of cloud computing solutions available for deep learning offered as a service by companies like Microzosft (Azure), Amazon (AWS), Google (AutoML Vision), and so on. These services enable developers and users with minimal deep learning expertise to train high-quality custom vision models (typically by leveraging the transfer learning technology) through a simple graphical user interface (GUI). One could also "rent GPUs in the Cloud" from these companies for training their deep learning models.

The predictive accuracy achieved by the deep learning based pavement distress detection has been so far impressive considering the challenges addressed by each of the reviewed studies. With relatively fewer instances of street view images, Some [31] achieved a very good accuracy of 0.9225 for detecting road surfaces with no cracks (as the overall goal was to reduce the street view images database for further processing by eliminating those road surfaces with no cracks). On the other hand, Ma et al. [45] achieved an average accuracy of 0.582 in predicting the pavement condition rating (taking into account the overall pavement condition and not just the cracks) using 700,000 street view images from 70,000 segments of New York City.

It is acknowledged that the summary of results presented in Table 3 may not present a fair comparison of the reported studies, primarily because different datasets were used by the studies. Unless the proposed methods are tested on a standard benchmarking dataset, it is difficult to make a fair comparison of their strengths and weaknesses, let alone recommending a winning strategy. Another complicating factor is the lack of a standard definition/consensus or metric for a pavement crack and the different types of cracks among the pavement and computer vision community.

Table 3. Overview of deep learning software framework, CPU and GPU specs, and summary test results from individual studies.

Reference	DL Software Framework	CPU Specs	GPU Specs	Test Results Summary	Method(s) for Comparison
Some [31]	DIGITS/Caffe	N/A	N/A	A = 0.9225; P = 0.9841; R = 0.8493	CrackIT
Zhang et al. [33]	Caffe	Intel® Xeon® E3-1241 V3 @ 3.5 GHz (8 GB RAM)	NVIDIA Quadro K220	P = 0.8696; R = 0.9251; F1 = 0.8965	SVM and Boosting Methods
Pauly et al. [34]	N/A	Intel® Xeon® E5-1630 v4 @ 3.70 GHz (128GB RAM)	NVIDIA Quadro M4000	A = 0.913; P = 0.907; R = 0.920	N/A
Eisenbach et al. [35]	Keras (Theano backend)	N/A	NVIDIA Titan X	A = 0.9772; F1 = 0.7246	CrackIT
Maeda et al. [36]	TensorFlow	N/A	NVIDIA GRID K520	AC: A = 0.85; P = 0.73; R = 0.68	N/A
Tong et al. [41]	Caffe	Intel® Core™ i7-6700 (8 GB RAM)	NVIDIA GeForce GTX	Recognition: A = 0.998	N/A
Wang and Hu [42]	TensorFlow	N/A	N/A	AC: A = 0.901	Neural Networks
Fan et al. [43]	TensorFlow	Intel® Xeon® E5-2690 2.9 GHz (64 GB RAM)	NVIDIA Quadro K5000	P = 0.9018; R = 0.9494; F1 = 0.9210	Canny; Local Thresholding; CrackForest
Ma et al. [45]	Keras (TensorFlow backend)	N/A	N/A	Average A = 0.582	SVM
Zhang et al. [21]	C++ (CUDA C Platform without using NVIDIA cuDNN library)	N/A	NVIDIA GeForce GTX Titan (2 devices)	P = 0.9013; R = 0.8763; F1 = 0.8886	Pixel-SVM; Shadow modeling
Gopalakrishnan et al. [47]	Keras (Theano backend)	Intel® Core™ i7-5600 @ 2.60 GHz (20 GB RAM)	Not used	A = 0.90; P = 0.90; R = 0.90; F1 = 0.90	N/A
Zhang et al. [51]	Caffe with MATLAB	HP Z220 workstation	NVIDIA Quadro K4000	P = 0.847; R = 0.951; F1 = 0.895	Canny; CrackIT; CrackForest

Note: N/A = not available or not applicable; SVM = support vector machine; A = accuracy; P = precision; R = recall; F1 = F_1 score; AC = alligator crack.

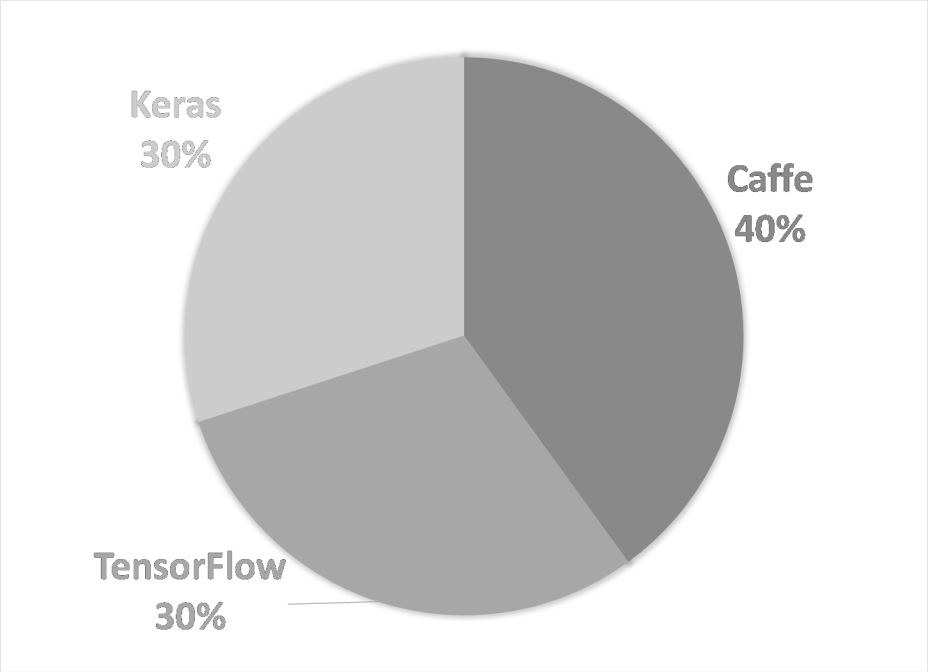

Figure 2. Distribution of deep learning software frameworks used by studies.

5. Summary, Conclusions, and Future Directions

Automated distress detection from pavement images is a challenging problem that has been actively investigated over the past three decades. The presence of shadows; inhomogeneity of distresses; lack of illumination and sufficient contrast between the distress and the background; presence of other objects such as patches and white lane markings, among others, are some major inherent challenges that are difficult to address. While the application of machine learning approaches to automated pavement distress detection is not new, the recent emergence and success of deep learning approaches in areas like speech recognition and computer vision has attracted the attention of pavement researchers. This paper reviewed some recent applications of deep learning to vision-based automated pavement distress detection.

Based on a small number of studies that have been reported so far in 2016–2018, it appears that interest in the use of deep learning in vision-based automated pavement distress detection is growing fast. Among the reviewed papers, two were published in 2016, seven in 2017, and three in 2018 (as of March). One of the articles included in the review is not yet published, but is uploaded to the arXiv repository. Most reviewed papers focused on crack detection from 2D pavement images or street view images as cracks are easier to detect compared with other 3D oriented distresses like rutting, pothole, and so on. As the state-of-the-practice in pavement distress data collection is moving towards 3D vision and measurement systems, more studies are expected to focus on analyzing 3D pavement images and move from overall image classification (crack or no_crack) to pixel-level crack detection [46].

Similar to other application areas, the convolutional neural networks (and variants) rank at the top in most pavement image classification studies. While some studies have shown that deeper CNNs achieve better performance than shallower networks, it is not clear whether the predictive accuracy can be improved by merely adding more layers to a CNN [34]. It has been demonstrated in image analysis competitions that the top performers are often distinguished from the rest by their ability to incorporate their domain expertise in some way into a deep learning pipeline along with the use of some data warping strategies [50]. Likewise, for automated pavement crack detection, data pre-processing and data augmentation at some level seems to be important for improving predictive accuracy.

While end-to-end deep learning is an attractive notion where one can go directly from raw data to desired result (e.x., a speech recognition task where one can directly go from raw audio recording to transcription) without the need for hand-crafted features or multi-step feature processing, it requires lots of input–output data for training efficient and accurate models. Strictly speaking, currently, there are no end-to-end deep learning models for vision-based automated pavement distress detection. This trend may change in the near future as more annotated or labeled pavement images dataset are being made publicly available [35,36]. Meanwhile, deep transfer learning appears to be a reasonable approach for researchers wanting to get familiarized with the application of CNNs to pavement distress detection, as pre-trained CNNs trained on "big data" natural image (annotated) datasets could be used on 2D pavement images as feature extractors [47,51].

Another major challenge is class imbalance and its effect on deep learning classification performance. Typically, pavement images containing distresses are fewer compared with those with no distress, especially considering a multi-class (longitudinal crack, transverse crack, alligator crack, etc.) problem. This disparity can be more pronounced in the context of street view images. This creates class imbalance, a problem that has been comprehensively studied in machine learning, but is beginning to be studied and addressed in the context of deep learning [55]. This could be an important area of future research in the context of deep learning application to automated pavement distress detection.

In terms of hyper-parameter optimization, there appears to be some overall consensus among the reviewed papers with respect to the use of ReLU activation function and the use of dropout to prevent over-fitting. Other parameters like the learning rate, weight decay, and so on, took on default values in most studies. This seems to suggest that fine tuning of hyper-parameters, although an important step, is only secondary to the selection of architecture and training data quality, which have far more impact on the network performance in vision-based pavement distress detection.

Apart from classification of distresses, characterizing the extent and severity of distresses are equally important in the context of a pavement management system and reporting requirements. This is an area that has been traditionally overlooked in the classical machine learning approach to automated pavement distress detection. Not surprisingly, the current deep learning studies reviewed in this paper did not address this topic, but it appears that most research groups are working on it as part of their ongoing research. This could once again be attributed to the lack of availability of large-scale annotated pavement image datasets that also include the extent and severity annotation.

Self-taught learning and unsupervised feature learning is a promising area of deep learning research, where the algorithm is enabled to learn from unlabeled data. A single unlabeled example is definitely less informative than a labeled one. However, if massive amounts of unlabeled examples are made available and if deep learning algorithms are designed to exploit this unlabeled big data effectively (by learning a good feature representation of the input), then they are expected to outperform traditional approaches requiring massive manual labeling [56]. As mentioned previously, public repositories with massive amounts of pavement images are being made available by researchers and organizations. Combined with this, crowd-sourced smartphone pavement images and recent explosion of interest in UAVs or drones for visual inspection, among others, are expected to result in massive amounts of unlabeled pavement images suitable for unsupervised feature learning. Variational auto-encoders (VAEs) and generative adversarial networks (GANs) are two innovative unsupervised strategies that could be explored further for end-to-end unsupervised deep learning of representative features from large amounts of unlabeled pavement images [50]. This will hopefully result in more robust deep learning solutions that are less sensitive to location variance (changes in locations and conditions of pavement images between the training and testing set).

The study of spatio-temporal variations in pavement surfaces is another area of research that can benefit from advances in deep learning application to pavement image analysis. Especially in the case of public image repositories that are routinely updated (e.x., Google StreetView images with timestamps), one could study the variations in texture and other surface characteristics with time [45].

This could also lead to research in the broader area of vision-based pavement anomaly detection, where one could study the evolution of distresses with time. Finally, recent advances in automated captioning of images using CNNs-recurrent neural networks (LSTMs) hybrid architectures, which combine the power of computer vision with natural language processing, may become a hot area of research with the overarching goal of achieving textural representation of pavement images describing the type, extent, and severity of distresses.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. ASCE. *American Society of Civil Engineers (ASCE) 2017 Infrastructure Report Card: Roads*; American Society of Civil Engineers (ASCE): Reston, VA, USA, 2017.
2. Flintsch, G.; McGhee, K. *NCHRP Synthesis 401: Quality Management of Pavement Condition Data Collection*; Transportation Research Board: Washington, DC, USA, 2009.
3. Gopalakrishnan, K. Advanced Pavement Health Monitoring and Management. IGI Global Videos, 2016. Available online: http://www.igi-global.com/video.aspx?ref=advanced-pavement-health-monitoring-management&titleid=137625 (accessed on 9 May 2017).
4. Tsai, Y.; Wang, Z. *Development of an Asphalt Pavement Raveling Detection Algorithm Using Emerging 3D Laser Technology and Macrotexture Analysis*; Final Report NCHRP IDEA Project 163; Transportation Research Board: Washington, DC, USA, 2015.
5. Wang, K.C.P.; Li, Q.J.; Yang, G.; Zhan, Y.; Qiu, Y. Network level pavement evaluation with 1 mm 3D survey system. *J. Traffic Transp. Eng. Engl. Ed.* **2015**, *2*, 391–398. [CrossRef]
6. Xie, D.; Zhang, L.; Bai, L. Deep Learning in Visual Computing and Signal Processing. *Appl. Comput. Intell. Soft Comput.* **2017**, *2017*, e1320780. [CrossRef]
7. Agrawal, A.; Choudhary, A. Perspective: Materials informatics and big data: Realization of the 'fourth paradigm' of science in materials science. *APL Mater.* **2016**, *4*, 053208. [CrossRef]
8. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
9. Bahrampour, S.; Ramakrishnan, N.; Schott, L.; Shah, M. Comparative Study of Deep Learning Software Frameworks. *arXiv*, 2016.
10. Fonnegra, R.D.; Blair, B.; Díaz, G.M. Performance comparison of deep learning frameworks in image classification problems using convolutional and recurrent networks. In Proceedings of the 2017 IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena, Colombia, 16–18 August 2017; pp. 1–6.
11. Shi, S.; Wang, Q.; Xu, P.; Chu, X. Benchmarking State-of-the-Art Deep Learning Software Tools. In Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD), Macau, China, 16–18 November 2016; pp. 99–104.
12. Caffe. Caffe—Deep Learning Framework. 2018. Available online: http://caffe.berkeleyvision.org/ (accessed on 15 February 2018).
13. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, New York, NY, USA, 19–21 August 1998; pp. 675–678.
14. TensorFlow. TensorFlow. 2018. Available online: https://www.tensorflow.org/ (accessed on 15 February 2018).
15. Theano. Theano at a Glance—Theano 1.0.0 Documentation. 2018. Available online: http://deeplearning.net/software/theano/introduction.html (accessed on 16 February 2018).
16. Torch. Torch—Scientific Computing for LuaJIT. 2018. Available online: http://torch.ch/ (accessed on 16 February 2018).
17. Chollet, F. *Keras*; GitHub: San Francisco, CA, USA, 2015.
18. Chambon, S.; Moliard, J.-M. Automatic Road Pavement Assessment with Image Processing: Review and Comparison. *Int. J. Geophys.* **2011**, *20*. [CrossRef]

19. Oliveira, H.; Correia, P.L. Automatic road crack segmentation using entropy and image dynamic thresholding. In Proceedings of the 2009 17th European Signal Processing Conference, Glasgow, UK, 24–28 August 2009; pp. 622–626.
20. Tsai, Y.-C.; Kaul, V.; Mersereau, R.M. Critical assessment of pavement distress segmentation methods. *J. Transp. Eng.* **2009**, *136*, 11–19. [CrossRef]
21. Zhang, D.; Li, Q.; Chen, Y.; Cao, M.; He, L.; Zhang, B. An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection. *Image Vis. Comput.* **2017**, *57*, 130–146. [CrossRef]
22. Ayenu-Prah, A.; Attoh-Okine, N. Evaluating Pavement Cracks with Bidimensional Empirical Mode Decomposition. *EURASIP J. Adv. Signal Process.* **2008**, *861701*, 2008. [CrossRef]
23. Subirats, P.; Dumoulin, J.; Legeay, V.; Barba, D. Automation of Pavement Surface Crack Detection using the Continuous Wavelet Transform. In Proceedings of the 2006 International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 3037–3040.
24. Wang, K.C.P.; Li, Q.; Gong, W. Wavelet-Based Pavement Distress Image Edge Detection with à Trous Algorithm. *Transp. Res. Rec. J. Transp. Res. Board* **2007**, *2024*, 73–81. [CrossRef]
25. Ying, L.; Salari, E. Beamlet Transform-Based Technique for Pavement Crack Detection and Classification. *Comput. Aided Civ. Infrastruct. Eng.* **2010**, *25*, 572–580. [CrossRef]
26. Koch, C.; Georgieva, K.; Kasireddy, V.; Akinci, B.; Fieguth, P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv. Eng. Inform.* **2015**, *29*, 196–210. [CrossRef]
27. Oliveira, H.; Correia, P.L. Automatic Road Crack Detection and Characterization. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 155–168. [CrossRef]
28. Fujita, Y.; Shimada, K.; Ichihara, M.; Hamamoto, Y. A method based on machine learning using hand-crafted features for crack detection from asphalt pavement surface images. In Proceedings of the Thirteenth International Conference on Quality Control by Artificial Vision 2017, Tokyo, Japan, 14 May 2017; Volume 10338, 103380M.
29. Hizukuri, A.; Nagata, T. Development of a classification method for a crack on a pavement surface images using machine learning. In Proceedings of the Thirteenth International Conference on Quality Control by Artificial Vision 2017, Tokyo, Japan, 14 May 2017; 103380M; Volume 10338, 103380M.
30. Zou, Q.; Cao, Y.; Li, Q.; Mao, Q.; Wang, S. CrackTree: Automatic crack detection from pavement images. *Pattern Recognit. Lett.* **2012**, *33*, 227–238. [CrossRef]
31. Some, L. *Automatic Image-Based Road Crack Detection Methods*; KTH Royal Institute of Technology: Stockholm, Sweden, 2016.
32. Oliveira, H.; Correia, P.L. CrackIT—An image processing toolbox for crack detection and characterization. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 798–802.
33. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
34. Pauly, L.; Peel, H.; Hogg, D.; Fuentes, R. Deeper Networks for Pavement Crack Detection. In Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC 2017), Taipei, Taiwan, 28 June–1 July 2017; pp. 1–7.
35. Eisenbach, M.; Stricker, R.; Seichter, D.; Amende, K.; Debes, K.; Sesselmann, M.; Ebersbach, D.; Stoeckert, U.; Gross, H.M. How to get pavement distress detection ready for deep learning? A systematic approach. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2039–2047.
36. Maeda, H.; Sekimoto, Y.; Seto, T.; Kashiyama, T.; Omata, H. Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone. *arXiv*, 2018.
37. Maeda, H.; Sekimoto, Y.; Seto, T. Lightweight Road Manager: Smartphone-based Automatic Determination of Road Damage Status by Deep Neural Network. In Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, New York, NY, USA, 31 October 2016; pp. 37–45.
38. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, Ch.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; Volume 9905, pp. 21–37.

39. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv*, 2017.
40. Dhakal, N.; Elseifi, M.A.; Zhang, Z. Mitigation strategies for reflection cracking in rehabilitated pavements—A synthesis. *Int. J. Pavement Res. Technol.* **2016**, *9*, 228–239. [CrossRef]
41. Tong, Z.; Gao, J.; Zhang, H. Recognition, location, measurement, and 3D reconstruction of concealed cracks using convolutional neural networks. *Constr. Build. Mater.* **2017**, *146*, 775–787. [CrossRef]
42. Wang, X.; Hu, Z. Grid-based pavement crack analysis using deep learning. In Proceedings of the 4th International Conference on Transportation Information and Safety (ICTIS), Banff, AB, Canada, 8–10 August 2017; pp. 917–924.
43. Fan, Z.; Wu, Y.; Li, W. Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network. *arXiv*, 2018.
44. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [CrossRef]
45. Ma, K.; Hoai, M.; Samaras, D. Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild. In Proceedings of the British Machine Vision Conference, London, UK, 4–7 September 2017.
46. Zhang, A.; Wang, K.C.P.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network. *Comput. Civ. Infrastruct. Eng.* **2017**, *32*, 805–819. [CrossRef]
47. Gopalakrishnan, K.; Khaitan, S.K.; Choudhary, A.; Agrawal, A. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr. Build. Mater.* **2017**, *157*, 322–330. [CrossRef]
48. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [CrossRef] [PubMed]
49. Bar, Y.; Diamant, I.; Wolf, L.; Lieberman, S.; Konen, E.; Greenspan, H. Chest pathology detection using deep learning with non-medical training. In Proceedings of the 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), New York, NY, USA, 16–19 April 2015; pp. 294–297.
50. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.W.M.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef] [PubMed]
51. Zhang, K.; Cheng, H.D.; Zhang, B. Unified Approach to Pavement Crack and Sealed Crack Detection Using Preclassification Based on Transfer Learning. *J. Comput. Civ. Eng.* **2018**, *32*, 04018001. [CrossRef]
52. Cha, Y.-J.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput. Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
53. Feng, C.; Liu, M.-Y.; Lee, T.-Y. Deep Active Learning for Civil Infrastructure Defect Detection and Classification. In Proceedings of the 2017 International Workshop on Computing in Civil Engineering, Seattle, WA, USA, 25–27 June 2017.
54. Gopalakrishnan, K.; Gholami, H.; Vidyadharan, A.; Choudhary, A.; Agrawal, A. Crack Damage Detection in Unmanned Aerial Vehicle Images of Civil Infrastructure Using Pre-Trained Deep Learning Model. *Int. J. Traffic Transp. Eng.* **2018**, *8*, 1–14.
55. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *arXiv*, 2017.
56. Stanford University. Unsupervised Feature Learning and Deep Learning Tutorial. Deep Learning Tutotrial, 2018. Available online: http://ufldl.stanford.edu/tutorial/selftaughtlearning/SelfTaughtLearning/ (accessed on 9 March 2018).