International Journal of Pavement Engineering

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gpav20

Evaluation of deep learning models for classification of asphalt pavement distresses

Alex Apeagyei, Toyosi Elijah Ademolake & Mark Adom-Asamoah

To cite this article: Alex Apeagyei, Toyosi Elijah Ademolake & Mark Adom-Asamoah (2023) Evaluation of deep learning models for classification of asphalt pavement distresses, International Journal of Pavement Engineering, 24:1, 2180641, DOI: 10.1080/10298436.2023.2180641

To link to this article: https://doi.org/10.1080/10298436.2023.2180641

Published online: 24 Feb 2023.

INTERNATIONAL JOURNAL OF PAVEMENT ENGINEERING
2023, VOL. 24, NO. 1, 2180641
https://doi.org/10.1080/10298436.2023.2180641

Evaluation of deep learning models for classification of asphalt pavement distresses

Alex Apeagyei[a], Toyosi Elijah Ademolake[a] and Mark Adom-Asamoah[b]

[a]School of Architecture, Computing and Engineering, University of East London, London, UK; [b]College of Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

ABSTRACT

Transfer learning (TL) offers a convenient methodology for exploiting the capability of deep convolutional neural networks (DCNNs) for many image classification tasks including the classification of pavement distresses. Seven state-of-the-art DCNNs were retrained to classify asphalt pavement distresses grouped into eight classes using TL techniques. The aim was to evaluate the predictive performances of the selected DCNNs in order to provide some guidelines on selection of DCNNs for pavement application. The results show some existing DCNN's are better than others for developing pavement distress classification models using the specific TL approach adopted in the study. The predictive ability of each model varied depending on distress class as some models with very low overall accuracy showed excellent results for individual distress class(s). Based on a combination of various performance metrics including F1-score, area under ROC curve, optimal operating threshold, training time, and model size, the best performing network had a relative score that was found to be significantly higher than the next two top-performing models. The best-performing networks were characterised by lower proportions of false negative values, low ambiguity scores, and well-defined t-SNE clusters that showed clear separation between the eight different pavement distress classes considered.

ARTICLE HISTORY
Received 16 June 2022
Accepted 10 February 2023

KEYWORDS
Asphalt; asphalt pavements; pavement distresses; pavement distresses classification; F1-score; transfer learning; rutting; fatigue cracking; transverse cracking; longitudinal cracking

1. Introduction

Most state transportation agencies conduct highway distress surveys to support their asset management systems. Accurately identifying flexible pavement distresses is of enormous importance to the management and maintenance of the highway systems. It serves as the basis for recommending remedial action(s) likely to result in the most cost-effective solution. Many attempts have been made to automate the process of collecting and identifying distress data using various visual media. Currently, the most widely used and reliable method of identifying pavement distresses involves manual and/or semi-automated data collection and analysis by well-trained technical personnel. In the UK, for instance, road surface conditions are often visually assessed either manually by a trained technician, or automatically, using vehicles equipped with lasers and cameras to measure various attributes of the road. While some aspects of the data collection have been automated (e.g. image or video data collection), classification of pavement distresses is still a tedious, manual and highly subjective task. Many existing methods for detecting and classifying pavement distress are semi-automated and limited to measuring particular aspects that must be analysed further by experienced technicians or expensive proprietary systems. Furthermore, as reported by Vavrik *et al.* (2013), existing automated systems can be expensive to acquire (e.g. $1.2 m/unit) and operate ($70k/year). Many existing automatic systems also tend not to be easy to use. As a result, even after acquiring these expensive systems, some highway agencies still rely on manual inspection by human experts as a more convenient solution owing to its ease of implementation (Siriborvornratanakul 2018). The traditional methods of pavement condition assessment, especially at the network level, rely almost exclusively on visual surveys of existing pavement distresses and in-situ assessments that are very subjective and often conducted intermittently. However, cost-effective pavement maintenance and rehabilitation solutions require continuous monitoring of distress initiation and propagation to assist in accurate timing of repairs. The inability of existing pavement condition assessment systems to realistically emulate the skills of highly trained road pavement technicians is a major unmet challenge that could be addressed by using deep learning techniques especially transfer learning of existing DCNNs. The main advantages of DCNNs over other machine learning algorithms for pavement distress classification include the following: (a) they outperform classical image classification methods in terms of accuracy; (b) DCNN architecture is flexible and thus could be adapted to new problems or to existing problems when new data become available, and (c) DCNN models are more robust because natural variations in data is automatically learned. Furthermore, compared to traditional pavement distress identification methods (such as digital image analysis), DCNN models have greater accuracy. Since neural networks used in DCNN are trained rather than programmed, applications using this approach often

CONTACT Alex Apeagyei a.apeagyei@uel.ac.uk

require less expert analysis and fine-tuning and utilise the vast amount of digital image data (video and still images) data currently available (Gopalakrishnan 2018, O'Mahony 2020). Additional advantages of DCNN is discussed in Section 2 for selected existing DCNNs. In spite of these advantages, as described in Section 2, several technological challenges need to be addressed before fully automated and practical TL-based DCNN models could be developed which warrants studies such as the current one.

DCNNs are a subset of artificial neural networks (ANNs) in which a multi-layered model learns to perform classification tasks directly from images, text, or sound, using neural network architecture; the more layers they have, the deeper the network. By comparison, traditional ANNs containing only 2 or 3 layers are considered 'shallower', while deep networks can have hundreds of layers (Bengio *et al*., 2013, 2015, Schmidhuber, 2015).

The development and application of DCNNs for general image classification have been demonstrated in several previous studies including Zeiler and Fergus (2013), Krizhevsky *et al*. (2012), Simonyan and Zisserman (2015) and Shin *et al*. (2016). The DCNN approach typically involves passing on an image through a sequence of convolutional layers and extracting detectable features within the convolved image. The convolved images are then classified using a fully connected neural network that is defined mathematically through a set of weights. Finally, the weights are optimised with a labelled training set through multinomial logistic regression using mini-batch gradient descent (Bottou 2010, Simonyan and Zisserman 2015).

The use of DCNNs for image classification has seen exponential growth since the early 2010s. Key factors that have contributed to the popularity and successful application of DCNNs for extensive image and video recognition include the availability of big public image repositories such as ImageNet (Deng *et al*. 2009), high-performance computing systems, such as GPUs or large-scale distributed clusters (Dean *et al*. 2012) and the public release of model architectures by individual research teams to the wider research community which, in turn, enables models to be retrained to classify new images using the concept of transfer learning. As a result, many highly accurate DCNNs classifiers have surfaced including Alexnet, Densenet201, Googlenet, Nasnetlarge, Resnet50, Squeezenet, and Xception for classifying many common natural and man-made objects.

For the current study, the transfer learning approach was used to retrain the above-mentioned DCNNs to classify asphalt pavement distresses. Transfer learning removes one of the major obstacles to the widespread application of DCNNs to the pavement engineering field – the lack of large-scale pavement distress datasets (on the order of millions of images). It should be noted that classification as used in this paper refers to labelling an image rather than an object in an image. The aim was to develop performance metrics that could be used for the selection of an automatic DCNN system that allows extension of its application to the pavement engineering field. A primary goal of this study was to evaluate selected, existing DCNN architectures, using a dataset of 400 images in classifying eight, common, asphalt pavement distresses using robust performance metrics.

The contributions of this study include the provision of comparative description of the architecture for each of the seven judiciously chosen DCNNs and the objective comparison, using robust performance metrics, of the behaviour of different DCNNs for future application to pavement distress classification. Furthermore, by identifying which of the selected networks best model pavement distresses and achieves good classification and validating it through multi-class confusion matrix statistical measures, the DCNN classifier can be implemented as an automated system for identification of common pavement distress thus contributing in no small way to the maintenance of the ageing highway pavement infrastructure. Finally, the approach presented focusing on size invariant measures which is well-suited for pavement distress datasets that are often imbalanced.

The rest of the paper is structured as follows: Section 2 presents an overview of related work. Section 3 describes the seven selected DCNNs architectures and the methodology used to conduct the experiments. The results and discussion are presented in Second 4. Finally, the conclusions and recommendations for future work are presented in Section 5.

2. Related work

The basic concept behind transfer learning (TL) is to utilise parts of a pre-trained DCNNs architecture for tasks in one domain for which a large amount of labelled data is available (such as the ImageNet dataset) in situations where only a small amount of labelled data is available (such as a pavement distress dataset). A key benefit of using TL is that adjusting the model parameters of a pre-trained network is much quicker, easier and requires a lesser number of labelled images than constructing and training a brand-new network from scratch. This is possible because the pre-trained network has previously been trained to recognise many valuable features from the large number of images (on the order of millions) on which it was trained in the first place. Two TL methods including feature transfer and parameter transfer are commonly used. In feature transfer, the last layer of a pre-trained network is removed, and its previous activation value sent to a classifier. In parameter transfer, only a few layers of the network need to be reinitialised with the remaining layers using the weight parameters of the pre-trained network and then using the new data set to fine-tune the network parameters. Fine-tuning as used here is a TL concept which involves replacing the pre-trained output layer with another layer containing the number of classes of the asphalt pavement distresses. Detailed description of the TL techniques for general machine learning applications has been provided in Pan and Yang (2010) to which interested readers may consult. In the following, the application of TL to pavement distress classification is presented. Most TL-based applications in the pavement distress classification domain are of recent origin with studies first reported in 2016 or later.

In one of the first studies using TL for pavement distress classification, Ma *et al*. (2017) leveraged publicly available geo-referenced road condition records along with Google Street Map images to develop a three-class (poor, fair and good) dataset comprising of 700 thousand images from 70

thousand street segments in New York, U.S.A. They used VGG-16D very deep convolutional neural network developed by Oxford University as the framework for training their network to classify pavement conditions. An average prediction accuracy of 58.2% was reported. The results highlighted the impact of class imbalance on the prediction accuracy as pavement classified as 'poor' formed only 0.6% of the total 711,520 images in the dataset.

Gopalakrishnan *et al.* (2017) used transfer learning with fine-tuning to retrain the VGG-16 DCNN model for automated pavement crack detection. A unique feature of the approach proposed by Gopalakrishnan *et al.* was that the VGG-16 model was used to vectorise the labelled pavement images, and a machine learning classifier was then used to predict the labels as 'crack' or 'no crack'. In a related study, Gopalakrishnan *et al.* (2018) used DCNNs with transfer learning for crack damage detection in unmanned aerial vehicle (UAV) images of civil infrastructure. The authors reported up to 90% accuracy in finding cracks in realistic situations without any data augmentation and pre-processing.

Maeda *et al.* (2018) retrained two existing DCNNs (Inception V2 and MobileNet) to classify various pavement surface features with eight class labels including five types of cracks, rutting-bump-pothole-separation, white line blur, and cross walk blur. The two DCNNs used were chosen based on their computational efficiency, lower CPU memory requirements, accuracy, and their potential to be deployed as smartphone apps for road damage detection in Japan. The predictive performances of the trained models showed overall accuracy (averaged over the eight data classes) was approximately 87% for the two models. The recalls varied depending on class label and model type and ranged from 0.03 to 0.81 for Inception V2, and 0.02 to 0.89 for MobileNet. The disparity in recall performance was attributed to class imbalance as distress class with low representations were associated with lower recalls and precision. For example, potholes which formed only 3% of the 15,435 images in the dataset received a recall rate of only 0.02 when using MobileNet. Recall, also known as sensitivity or true positive rate, is the fraction of all positive samples that are correctly predicted as positive by the classifier; therefore, the model could only identify 2% of all potholes used in the study.

Nie and Wang (2018) used the fine-tuned transfer learning method consisting of limited initialisation and the Resnet50 DCNN architecture to classify pavement distresses grouped into four classes including 'crack', 'loose', 'deformation' and 'others'. The four classes comprised of multiple subclasses including: crack (crack, block crack, longitudinal crack and transverse crack), loose (groove, loose), deformation (subsidence, rutting, wave gushing) and other categories (flooding, repairing, frost heave and frosting). It should be noted that the class labels used appear quite unusual in the pavement field as it is often desirable to distinguish between the various types of cracking like longitudinal versus transverse in order to accurate identify the cause(s) of the distress. Furthermore, class distribution of the distresses was not reported. Nie and Wang reported overall average classification accuracy of the model as 96.53% with individual F1-scores of 97.73%, 88.59%, 88.57% and 81.1%, respectively for crack, loose, deformation and others. Nie and Wang noted that their model was limited to single distress images with simple background and recommended that that for practical applications, models that can classify multiple distresses within the same images may be needed.

Majidifard *et al.* (2020) collected approximately 7000 Google Street images and manually annotated them using nine pavement distress class labels to create one of the most comprehensive datasets in the field. The nine distresses included reflective crack, transverse crack, block crack, longitudinal crack, alligator crack, sealed reflective crack, lane longitudinal crack, sealed longitudinal crack, and pothole. A large class imbalance was indicated (e.g. potholes formed approximately 1% of all labelled distresses). The main motivation for the study was to demonstrate how the wide-view images obtained from Google Street could be used along with a deep learning approach to classify pavement distresses. Two DCNNs including You Only Look Once version 2(YOLO v2) and Faster Region Convolution Neural Network (Faster R-CNN) were retrained to classify pavement distresses. Based on F1-scores of 0.84 and 0.65 for YOLOv2 and faster F-CNN, respectively, the authors considered the results acceptable considering the convenience of utilising Google maps images.

Peraka *et al.* (2021) proposed a transfer learning approach using the You Only Look Once version 4 (YOLO v4) architecture to quantify multiple asphalt pavement distress types and severity levels. Their model achieved average precision 87.44% after 7,900 iterations.

Chen *et al.* (2022) applied a ten-fold cross-validation training method using EfficientNet B4 as the DCNN architecture to detect nine pavement features including alligator cracks, joint or patches, longitudinal cracks, manholes, oil stains, potholes, road markings, shadows, and transverse cracks. The unique feature of this study is the use of thermal imaging camera that enabled a trio of distress images (RGB, thermal and fusion of RGB and thermal) to be used in the training of the chosen DCNN – EfficientNet B4. Unlike most existing studies, a balanced data set comprising of 500 RGB and 500 thermal images for each of the nine pavement features was used. The results showed that the best predictions were obtained when the fused images (RGB plus 50% thermal) were used to develop the classification models with F1-score and recall of 98.34%. Similar to other studies, the developed models struggled to distinguish between longitudinal cracks and transverse cracks.

However, these existing studies provide limited information on the effect that either the architecture of the models themselves or the hyper-parameters, whose values control the training and learning process, have on the task of pavement distress identification. Previous studies have shown that performances achieved by different architectures and hyper-parameters on the same task vary and are non-universal, even those that have shown excellent performance on the industry standard, ImageNet (Deng *et al.* 2009). A review of available literature also suggests that very few comparative studies of DCNN architectures or methods have been performed using commonly available pavement distress images. Thus, guidance on the selection of DCNN architectures that are suitable for identifying asphalt pavement distress is lacking. A summary of various transfer learning studies including the approaches, model complexities, methodologies and key performance parameters used are summarised in Table 1.

Table 1. Overview of TL-based DCNN models for pavement distress classification.

Reference	DCNN architectures used	Key features including reported performance metrics
Ma *et al.* (2017)	VGG-16D (Size on disk – 528 MB, trainable parameters – 138 million)	Three class labels (poor, fair, good); Class imbalance identified; low recalls attributed to class imbalance; Overall accuracy of 58.2%
Gopalakrisna *et al.* (2017)	VGG-16D (Size on disk – 528 MB, trainable parameters – 138 million)	Two classes (crack, no crack); accuracy 90%
Gopalakrisna *et al.* (2018)	VGG-16 (Size on disk – 528 MB, trainable parameters – 138 million)	Accuracy (89%); precision (91%); recall (89%); F1-score (89%); Cohen's Kappa score (79%); AUC (0.9)
Maeda *et al.* (2018)	Inception V2 (Size on disk – 92 MB, trainable parameters – 23.9 million), Mobilenet (Size on disk – 16 MB, trainable parameters – 4.3 million)	Eight class labels (D00,Do1, D10, D11, D20, D40, D43,D44); class imbalance identified; recall varies depending on class label size; recall; precision; speed of training; accuracy (87%); recall (71%); precision (77%)
Mandal *et al.* (2018)	YOLO v2, YOLO v3, Faster R-CNN, single shot multibox detector (SSD)	Eight class labels (D00, D01, D10, D11, D20, D40, D43,D44); class imbalance; precision (77%); recall (73%); F1-score (75%)
Nie and Wang (2018)	Resnet50 (Size on disk – 98 MB, trainable parameters – 25.6 million)	Four class labels (crack, loose, deform, others); overall accuracy 96.5%
Majidifard *et al.* (2020)	YOLO v2, Faster R-CNN (size and parameters depends on backbone networks)	Nine class labels (reflective crack, transverse crack, block crack, longitudinal crack, fatigue crack, sealed reflective crack, lane longitudinal crack, sealed longitudinal crack, pothole); class imbalance; precision (93%), recall (77%), and F1-score (84)%
Peraka *et al.* (2021)	YOLO v4 (size and parameters depends on backbone networks)	Eighteen distress classes consisting of three levels of severity pertinent to cracking, potholes, and patch deterioration; average precision 87.44%
Ranjbar *et al.* (2021)	AlexNet, GoogleNet, SqueezNet, ResNet-18, ResNet-50, ResNet-101, DenseNet-201, and Inception-v3 (size 92 MB, parameters 189 million)	Four distress classes:Linear-cracking, Non-cracking, Surface-cracking and General. Similar performance (in terms accuracy, sensitivity, etc.) were reported. For example, accuracy for all modeles ranged from 0.965 to 1.
Chen *et al.* (2022)	Efficientnet eight versions (B0–B7). Model size and parameters varies depending on version: Size – 31–166 MB and number of parameters 186–438 million	Nine class labels (transverse cracks, longitudinal cracks, fatigue cracks, joint or patches, potholes, manholes, shadows, road markings and oil stains); accuracy (97.3%), precision (97.2%), recall (96.9%), F1-score (97.0%)
Zhu *et al.* (2022)	Faster R-CNN with ResNet50 and VGG16 as backbones; YOLOv3 and YOLOv4 with DarkNet53 and CSPDarkNet53 as backbone	Six pavement distress types labels (transverse crack, longitudinal crack, alligator crack, oblique crack, pothole, and repair); average accuracy ranged from 33% to 77%; mean average precision ranged from 38.7% to 56.6%. YOLO-based models performed relatively better than faster R-CNN-based models.

Notes: D00 = Longitudinal crack – wheelpath; D01 = Longitudinal crack – Construction joint; D10 = Transverse crack – Equal interval; D11 = Transverse crack – Construction joint; D20 = Fatigue crack; D40 = Rutting, bump, pothole, or separation; D43 = White line blur; D44 = Cross walk blur.

The review of existing studies shows that majority of transfer-learning based studies on pavement distress classification were limited to the evaluation of one or two existing models. Furthermore, different datasets used makes generalisation of the developed TL models difficulty. Additionally, some existing models do not transfer accurately when applied to new learning so the use of one or two models for training is a major limitation for the pavement distress identification area. The number and definition of distress classes varied widely from 2 to about 9. Finally, none of the reviewed studies used composite performance measures such as ROC, AUC, t-SNE, which have been shown to be more robust than more common measures like accuracy, or F1-score or precision, which were predominantly used in the reviewed TL models. Based on the related works that were reviewed, it was decided to limit the trained DCNNs to the following seven state-of-the-art DCNNs including Alexnet, Densenet201, Googlenet, Nasnetlarge, Resnet50, Squeezenet, and Xception. Even though some of selected modelshave been used in previous TL application for pavement distress identification, very few, if any, haveused the more robust graphical performance measures such as ROC, AUC, t-SNE and ambiguity.

3. Materials and methods

The main steps used to accomplish the study including acquisition of pavement distress data, fine-tuning of selected models with the procured data, data classification and data evaluation are presented next with justifications. Subsequently, after describing the rational for selecting the seven DCNN models, the section concludes with brief descriptions of the architecture and key operational parameters of each model.

3.1. Data acquisition

Approximately 400 freely available images of pavement distresses were acquired from multiple publicly available sources including Google Street, which is a common approach in the field. The 400 images were categorised manually into eight class labels, namely: block cracking, distress free, fatigue cracking, longitudinal cracking, patching, pothole, rutting, and transverse cracking. For each classified image and network type, user-defined functions were used for pre-processing into the required input size, as shown in Table 2. The images

Table 2. Lists of selected, deep convolutional neural networks and their properties.

Network	Depth	Size	Parameters (millions)	Image input size	Top-1 accuracy (%)
Alexnet	25	227 MB	61.00	227-by-227	55.9
Squeezenet	68	0.52 MB	1.24	227-by-227	57.5
Googlenet	144	27 MB	7.00	224-by-224	69.8
Densenet201	708	77 MB	20.00	224-by-224	77.3
Resnet50	177	96 MB	25.60	224-by-224	74.9
Xception	170	85 MB	22.90	299-by-299	79.0
Nasnetlarge	1243	332 MB	88.90	331-by-331	82.5

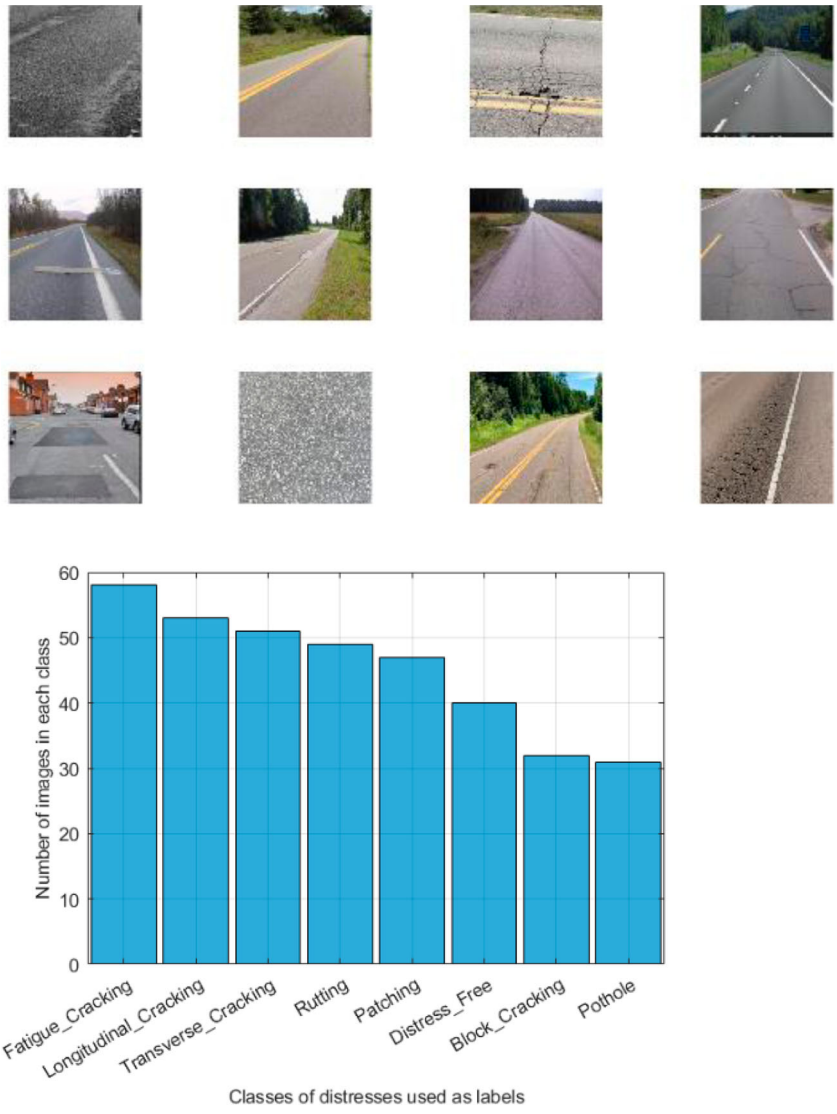

Figure 1. Sample images used in the training and validation of networks (left) and the sizes of each distress class (right).

were randomly grouped into training (85%) and validation (15%). Also shown in Table 2 are the reported performance metric (top-1 accuracy) which was one of the main criteria used in the selection of the models for inclusion in the study in the first place. Figure 1 (left) shows samples of images used in this study, while Figure 1 (right) shows the distribution of images in each distress class. As can be seen in Figure 1 (right), the distribution is imbalanced, with the number of images per class ranging between 30 and 60. Such imbalanced or skewed class distributions is common with most pavement distress datasets. As noted previously, using single-parameter performance measures such as precision, accuracy or recall, as used in the reviewed studies, may lead to inaccurate evaluation results.

3.2. Fine-tuning of selected DCNNs

As indicated previously, the seven existing, DCNNs selected for evaluation included Alexnet, Densenet201, Googlenet, Nasnetlarge, Resnet50, Squeezenet, and Xception (Table 2). The initial selection criteria included speed of training, accuracy, and model size. Each network shown was trained by machine learning experts on a subset of ImageNet which contains millions of images of common items grouped into 1000 object categories (e.g. keyboard, mouse, pencil, monkeys, etc.). Brief descriptions of the selected networks in Table 2. For this study, the fine-tuning tasks for each model included replacing their last three layers including the fully connected layer, the softmax layer, and the classification layer. Thus, for each network, all layers except the last three were frozen. This is a common approach used by previous researchers but it must be noted this is not universal. Even though the selection of the total number of layers to freeze or to train varies among different investigators, the main objective in each case is usually to achieve the best possible predictive accuracy for their models. Some existing TL works involve replacing more layers than the three aforementioned layers used in the current study but it must be noted that in general replacing more layers do not always lead to better retrained models.

The key characteristics of the seven models selected for pavement distress classification is presented below. Interested readers are referred to Wu *et al.* (2016) who have provided

detailed reviews of common existing DCNNs for image recognition.

3.2.1. Alexnet

Alexnet, which was proposed by Krizhevsky *et al.* (2012), won first prize in the ImageNet Large Scale Visual Recognition Challenge image classification contest in 2012 and set a major precedent for the field of deep learning. Several variants of DCNNs have since been developed as improvements on the original Alexnet architecture to enhance further the performance of DCNNs on image identification tasks. They include Squeezenet (Iandola *et al.* 2016), Googlenet (Szegedy *et al.* 2015a), and Resnet (He *et al.* 2016), to name just a few.

The basic architecture of Alexnet comprises of convolutional layers, normalisation layers, pooling layers, and fully connected layers. There are five convolutional layers, three max-pooling layers, two normalisation layers, two fully connected layers, and one softmax classifier layer. Compared with previous neural networks, two precedent setting components introduced in Alexnet include replacing the sigmoidal activation function (tanh units) with Rectified Linear Units (ReLUs) to make the training process faster and introducing dropout to reduce over-fitting. The first element of the layers property of the network is the image input layer, which requires image input of size $224 \times 224 \times 3$ with 'zero-centre' normalisation, where 3 is the number of colour channels. The first convolutional layer uses $11 \times 11 \times 3$ filters with stride [4 4] and padding [0 0 0 0]. Alexnet achieved top-1 accuracy of 55.9% to win 1st place on the ILSVRC 2012 classification task, setting the stage for the rapid development of DCNN for image classification.

3.2.2. Densenet201

Densely connected convolutional networks (Densenets), similar to Googlenet, were introduced by Huang *et al.* (2018) to address the vanishing gradient descent and other problems associated with very deep neural networks. Densenets simplify the connectivity pattern between layers by simply connecting every layer directly with each other compared with a traditional convolutional neural network (CNN) like Alexnet in which the output from a previous layer serves as the input to the next layer. This approach ensures maximum information (and gradient) flow, as well as requiring fewer parameters than an equivalent, traditional CNN, as learning superfluous feature maps becomes unnecessary. Densenets enhance the ability of the network through feature re-use instead of drawing representational power of the network to assign proper labels and create well-defined decision boundaries from extremely deep or wide architectures, as implemented in other networks such as Resnets. Densenets are considered to be 'dense' because every layer adds 32 new feature maps to the previous volume, going from 64 to 256 after 6 layers in Densenets. Densenet uses $224 \times 224 \times 3$ images with 'z-score' normalisation. The first convolutional layer in Densenet201 uses $7 \times 7 \times 3$ convolutions with stride [2 2] and padding [3 3 3 3]. Only one convolutional layer is fully connected, as opposed to two in Alexnet.

3.2.3. Googlenet

To address the problem of over-fitting and exploding or vanishing gradients commonly associated with networks with deep layers, Szegedy and co-workers introduced the Inception framework comprising of filters with several sizes that can operate on the same level (Szegedy *et al.* 2015a, 2015b, 2017). One consequence of the vanishing gradient descent problem is that the network stops learning during training. By utilising auxiliary classifiers in the course of training and removal them during inference, Inceptions models are better positioned to prevent over-fitting.

The Inception architecture results in increases in both the depth and the width of CNN while maintaining an affordable computational cost. Googlenet is the name given to the most common, Inception-based, deep learning network. Googlenet contains of nine Inception modules, four convolutional layers, four max-pooling layers, three average pooling layers, five fully connected layers, and three softmax layers for the main auxiliary layers. Two max-pooling layers are used between some inception modules, ReLu activation is applied in all of the convolutional layers and dropout regularisation is used in the fully connected layers.

Max-pooling layers is used in a DCNN to reduce the dimensions (i.e. height and width) of the input image as it is moved through the model. The first convolution layer in Googlenet uses a filter (patch) size of 7×7, which is similar to Alexnet. The main goal of the first layer is to reduce the input image immediately, but not lose spatial information by using large filter sizes. Googlenet uses $224 \times 224 \times 3$ images with 'zero-centre' normalisation and one fully connected layer. It achieved top-1 accuracy of 69.8% to win 1st place on the ILSVRC 2014 classification task.

3.2.4. Nasnetlarge

Zoph *et al.* (2018) developed Nasnetlarge, a Neural Architecture Search (NAS) model while working under the auspices of Google. Its development was inspired by the concept of reinforcement learning in which the search for a neural network architecture for training is based on giving rewards for desired behaviours and/or penalising undesired ones (Zoph and Le 2017). The approach requires enormous computing power and ingenuity to search for the best combination of model parameters such as filter sizes, output channels, strides, number of layers, etc., for automatic design of neural networks. Nasnetlarge is a pre-trained model that has been trained on a subset of the ImageNet database, similar to the six networks described previously. It uses $3 \times 3 \times 3$ convolutions with stride [2 2] and padding [0 0 0 0]. The first element of the layers property of the network is the image input layer, which requires image input of size $331 \times 331 \times 3$ with 're-scale-symmetric' normalisation. The authors reported top-1 accuracy of 82.7%, and top-5 accuracy of 96.2% when used on ImageNet dataset.

3.2.5. Resnet50

Deep neural networks can be difficult to train and can suffer from vanishing gradient problems, as previously discussed. Residual networks (Resnets) were introduced as improvements to address some of these issues (He *et al.* 2016). ResNets

achieve this improvement by adding skip connections, or short-cuts, parallel to the layers of convolutional neural networks, to jump over some layers. The skip connections in Resnets resolve the vanishing gradient problems of DCNNs in two ways: (1) by permitting alternative short-cut routes through which the gradient can flow, and (2) by allowing the model to learn the identity functions which ensures that the higher layer will perform at least as well as the lower layer, and not worse. There are several variants of Resnets, including Resnet18, Resnet50, Resnet101, Resnet152 and Resnet1000. For this study, Resnet50 was selected. As the name suggests, Resnet50 is a residual network with 50 residual blocks. Resnet50 is a comparatively deeper network with 177 layers. It came first in the 2015 version of the ImageNet Challenge (He *et al*. 2016), where it achieved top-1 accuracy of 74.9% in the classification task. It uses $7 \times 7 \times 3$ convolutions with stride [2 2] and padding [3 3 3 3] and input image size of $224 \times 224 \times 3$.

3.2.6. Squeezenet

Squeezenet was developed by Iandola *et al*. (2016) as a model with Alexnet-level accuracy but with a relatively smaller architecture. Compared with Alexnet, Squeezenet's architecture has a relatively smaller footprint in terms of file size and number of parameters.

By employing model compression techniques, Iandola *et al*. (2016) were able to compress Squeezenet to less than 0.5MB, which is 510 times smaller than Alexnet. Squeezenet has accuracy comparable with Alexnet but with only 1.24 million parameters, which is approximately 50 times less parameters than Alexnet. Strategies used in Squeezenet to lessen the number of required parameters include replacing the 3×3 max-pooling filters with 1×1 filters, decreasing the number of input channels to 3×3 filters, and down sampling later in the network. Squeezenet uses image input of size $227 \times 227 \times 3$ images with 'zero-centre' normalisation, similar to Alexnet. However, unlike Alexnet, no fully connected layers are present in the Squeezenet architecture. Compared with Alexnet, Squeezenet has a much deeper architecture with 68 total layers instead of Alexnet's 25. The first convolutional layer uses $3 \times 3 \times 3$ convolutions with stride [2 2] and padding [0 0 0 0].

3.2.7. Xception

Xception (for 'Extreme Inception') is a DCNN architecture where Inception modules have been replaced with depth-wise separable convolutions. Compared to spatial separable convolutions, depth-wise separable convolutions work with kernels that cannot be 'factored' into two smaller kernels. Xception has been described as an Inception model with many towers or as a linear stack of depth-wise, separable convolution layers with residual connections that uses 36 convolutional layers as a basis for feature extraction (Chollet 2016). In the Xception architecture, there is no ReLU non-linearity but point-wise convolution is followed by depth-wise convolution. Xception uses $3 \times 3 \times 3$ convolutions with stride [2 2] and padding [0 0 0 0]. On the ImageNet validation dataset, the model achieved a top-1 accuracy of 79% and top-5 accuracy of 94.5%.

3.2.8. DCNNs model settings and hyperparameters

At the fundamental level, the purpose of a DCNN is to transform images into data from which useful features for prediction can be extracted. The typical DCNN architecture consists of an input layer and multiple hidden layers (Figure 2). The hidden layers typically comprise of convolutional, pooling and flattening and fully connected layers, and an output or classification layer; and are often activated by rectified linear unit (ReLU). Out of this basic structure, and with some converging or modification, comes the many different DCNNs including the seven selected for the current study. Using a combination of multiple convolutional layers and pooling layers, a given image is processed for feature extraction. At the end of the convolutional and pooling layers, are a set of fully connected layers leading to the layer for SoftMax (for a multi-class case) or sigmoid (for a binary case) function.

A DCNN can have tens or hundreds of layers, with each layer learning to detect different features. The output of each convolved image is used as the input to the next layer. The filters or kernels initially detects very simple features, such as brightness or edges. More complex features that uniquely define the object are detected with deeper layers. Both the ReLU and the pooling layers are used to improve computational efficiency. By maintaining positive values while mapping negative values to zero, the ReLU permits quicker and more effective training. A pooling layer works by performing non-linear down sampling and reducing the number of parameters that the network needs to learn. Flattening layers converts the network's 2-dimensional spatial features into 1D vector of image-level features for image classification purposes. SoftMax provides probabilities for each category in the dataset (Bengio *et al*. 2013, 2015, Schmidhuber 2015).

Stochastic gradient descent method was used as the optimisation routine for TL of the selected networks because of its accuracy and efficiency. Similar hyper-parameters were selected for use to ensure that realistic comparisons among different DCNN architectures. In a DCNN, hyperparameters are used to control the learning process that determine model parameters that a network eventually learns. For this study, the model hyperparameters used included the following: (1) initial learning rate of 0.001, momentum of 0.9, L2 regularisation of 0.0001, epochs of 12 and minibatch size of 5.

A graphical processing unit (GPU) with an NVIDIA® T1000 based on Turing architecture, and an Intel Core i-7 CPU at 2.6 GHZ operating on a Windows 10 Pro 64-bit operating system were used. A common mini-batch size of 8 was used for all the DCNNs except Nasnetlarge where a mini-batch size of 1 was used because of memory limitations. Mini-batches are samples of the training dataset that are processed on the GPU at the same time and therefore can impact the speed of training and the accuracy of a network. The larger the mini-batch, the faster the training. However, larger mini-batch sizes are associated with longer training times. The networks were compiled using the stochastic gradient descent (SGD) optimisation technique. To fine-tune the selected models for the transfer learning process for each model, the last fully connected layer of the original network was replaced with a new fully connected layer, which classified the features into the eight pavement distress categories.

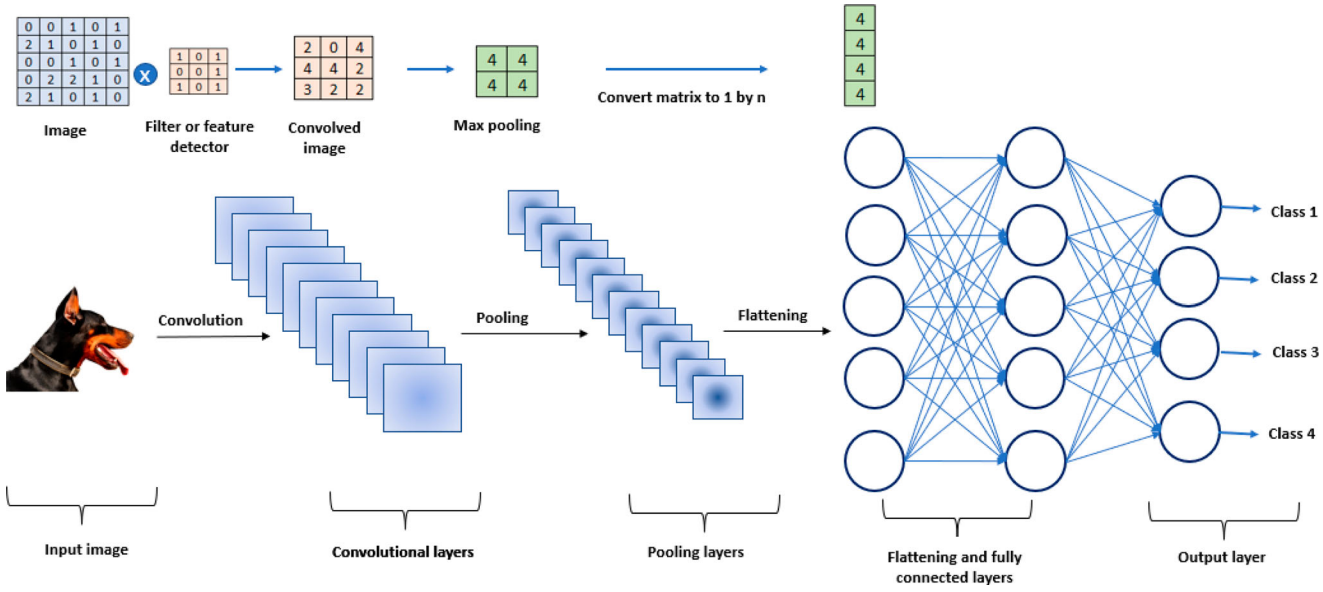

Figure 2. Basic architecture of a DCNN. In this example an image of a dog is transformed into a matrix of integers and a kernel is applied. The resulting convolved image is max pooled to produce a feature map which is then fed through flattening layers and finally a classification layer.

Each network was retrained to identify eight categories of flexible pavement distresses. The steps used to accomplish the transfer training of each network included: (1) importing the pre-trained network, (2) configuring selected layers to perform a new recognition task, (3) training the network on a pre-processed pavement distress dataset and (4) testing the results to predict and assess network accuracy. A schematic of the process is shown in Figure 3.

3.3. Evaluation of retrained models

The prediction performance of each retrained DCNN model was evaluated by comparing single confusion matrix statistics and accuracy measures such as precision, overall accuracy, recall and sensitivity which are commonly used to assess how well TL-based DCNNs perform by most previous investigators. In addition, combined measures such as F1-score and graphical measures including ROC, AUC, t-SNE, etc., that are more robust against class imbalance were used.

3.3.1. Confusion matrix

A confusion matrix (CM) can be applied to visualise the predictive performance of a DCNN model in a tabular fashion. Each element in a CM represents the number of predictions made by the network and whether it classified the classes correctly or wrongly. The sum total of the diagonal entries of a CM is used commonly to evaluate the success or otherwise of a DCNN classifier. To interpret the confusion matrix correctly, a few fundamental concepts should be considered. Interested readers are referred to Düntsch and Gediga (2019). For the simple case of a two-class classification problem, it is necessary to classify only two classes (typically a positive and a negative class). In such a problem, four metrics are typically used, including true positive (t_p), false positive (f_p), true negative (t_n), and false negative (f_n). A t_p denotes the number of predictions where the network correctly predicts the positive class as positive, while a t_n refers to the number of predictions where the network correctly predicts the negative class as negative. On the other hand, an f_p represents the number of predictions where the network incorrectly predicts the negative class as positive, while an f_n denotes the number of predictions where the network incorrectly predicts the positive class as negative. Even though the four metrics were originally specified for a binary classification problem, they are easily extended to classification problems that involve multiple classes. For a given network, the metrics can be used to estimate key performance measures such as accuracy, F1-score, precision, recall, specificity, Matthews correlation coefficient, receiver operating characteristic curve (ROC) and area under the curve (AUC). In addition to the aforementioned CM measures, the DCNN were evaluated using ambiguity of a classification and t-distributed stochastic neighbour embedding (t-SNE) function.

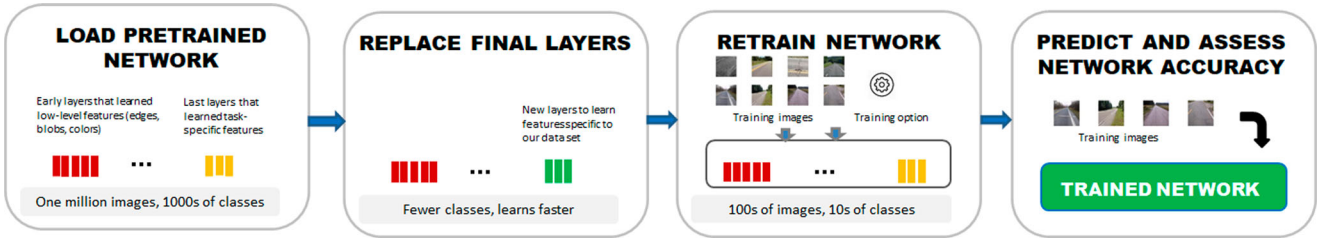

Figure 3. Schematic of transfer learning technique for training of selected deep neural networks to classify flexible pavement distresses.

Accuracy is a measure of the overall accuracy of DCNN and is defined as the proportion of the total samples that are correctly categorised by the classifier (Equation (1) below). In this study, accuracy refers to the overall accuracy of a model in classifying the eight classes of pavement conditions (block cracking, distress free, fatigue cracking, longitudinal cracking, patching, pothole, rutting and transverse cracking). Precision denotes what proportion of predictions assigned to a positive class are actually positive (Equation (2) below). Recall, also known as sensitivity or true positive rate, is the portion of all positive samples that are correctly predicted as positive by the classifier. Recall of a classifier can be calculated using Equation (3). Specificity or true negative rate (Equation (4) below) represents the proportion of all negative samples that are correctly predicted as negative by the network.

Overall accuracy is evaluated by the F1-score defined as the harmonic mean of the recall values and the precision (Equation (5)). The F1-score metric represents the compromise between precision and sensitivity. An F1-score is 0 when either the precision or the sensitivity is 0. For certain classification problems (e.g. unbalanced-class problems), Matthews correlation coefficient (MCC) is recommended (Equation (6)) as it considered that MCC is least affected by unbalanced data. It represents a correlation coefficient between the measured and predicted classifications and has values that range from −1 to +1, where a value of +1 represents a perfect prediction, 0 is no better than random prediction, and −1 represents the worst possible prediction (Akosa 2017).

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{1}$$

$$\text{Precision} = \frac{t_p}{t_p + f_p} \tag{2}$$

$$\text{Sensitivity} = \frac{t_p}{t_p + f_n} \tag{3}$$

$$\text{Specificity} = \frac{t_n}{t_n + f_p} \tag{4}$$

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2t_p}{2t_p + f_p + f_n} \tag{5}$$

$$\text{MCC} = \frac{t_p * t_n - f_p f_n}{\sqrt{(t_p + f_p) * (t_p + f_n) * (t_n + f_p) * (t_n + f_n)}} \tag{6}$$

3.3.2. Receiver operating characteristic (ROC) curves

A receiver operating characteristic curve (ROC) is a true positive rate (TPR) versus false positive rate (FPR) plot that can be used to display the performance of a network at all classification thresholds. It is considered as one of the most robust measures of the predictive performance of a DCNN classifier. The magnitude of the classification threshold controls the number of items classified as positive. Thus, a network operating at lower classification thresholds will classify more items as positive than a network operating at a higher threshold. An ROC can be used to determine other useful performance metrics including the optimal operating classification threshold (OPROCPT) and the areas under the ROC curve (AUC). Both OPROCPT and AUC have values between 0 and 1, with values closer to 1 associated with better performance.

3.3.3. Area under the ROC curve (AUC)

The area under the ROC curve or AUC is a measure of the area with coordinates ranging from (0,0) to (1,1) and therefore has a magnitude of 1. A model with an AUC of 0 will be expected to make predictions that will be 100% wrong. A model with an AUC of 1.0 will be expected to be correct 100% of the time and will rank all positives higher than all negatives. In practice, it is expected that reliable classification model will rank a random positive example higher than a random negative example more than 50% of the time and have an AUC in the range 0.5–1.0.

AUC is considered a more robust measure of performance than some of previously reviewed measures such as accuracy, F1-score and recall as it is not affected by class imbalance. This because AUC is considered to be scale-invariant as it evaluates how predictions are ranked rather than the absolute values of prediction levels. Another reason AUC is considered a more robust prediction performance measure is that it evaluates the predictive quality of a model at all possible threshold values, i.e. it is threshold-invariant.

3.3.4. Ambiguity of classification

A common problem of image identification and classification based on the deep convolutional neural network (DCNN) technique is that the rationale for the output judgement is often considered to be unclear. The problem could be addressed using the ambiguity parameter, where ambiguity of a classification is defined as the ratio of the second-largest probability to the largest probability (van der Maaten and Hinton 2008). The ambiguity of a classification ranges between zero (nearly certain classification) and 1 (likely to be classified to the most likely class as the second class). An ambiguity of near 1 means the network is unsure of the class in which a particular image belongs. This uncertainty might be caused by two classes whose observations appear so similar to the network that it cannot learn the differences between them. On the other hand, a high ambiguity can occur because a particular observation contains elements of more than one class, so the network cannot decide which classification is correct. It is noted that low ambiguity does not necessarily imply correct classification; even if the network has a high probability for a class, the classification can still be incorrect. For this study, the SoftMax activations were used to calculate the image classifications that were most likely to be incorrect. As can be seen in Table 1, none of the existing works reviewed reported this important performance evaluation parameter.

3.3.5. Data visualisation techniques

To understand better how a DCNN works to isolate pavement distress images into clusters depending on their physical features, the t-distributed stochastic neighbour embedding (t-SNE) function was used on the test images to view activations in a trained network. The t-SNE approach was by proposed by van der Maaten and Hinton (2008) as a non-linear dimension reduction technique for mapping high-dimensional data, such as network activations in a layer, into two dimensions. For the current

study, the t-SNE function was used to reduce the multi-dimensional activations of the SoftMax layer to a 2D representation with a similar structure. Tight clusters in the resulting t-SNE plot correspond to classes that the network usually classifies correctly. The visualisation thus made it possible to evaluate readily which observation that the network misclassified.

4. Results and discussion

4.1. Confusion matrix

Sample confusion matrices summarising predictive performances for the top performing (retrained Googlenet) and least performing (retrained NasnetLarge) networks are shown in Figure 4. In Figure 4, the rows relate to the predicted class (Output Class) and the columns correspond to the true class (Target Class). The observations that were correctly classified are depicted on the diagonal cells while the off-diagonal cells relate to wrongly classified observations. The number of observations is shown in each cell.

The two columns on the far right of the plot shows the proportions of all the distresses predicted to go to each class that are accurately classified (precision or positive predictive value) and wrongly classified (false discovery rate). The two rows at the bottom of the plot shows the percentages of all the distresses belonging to each class that are correctly classified (true positive

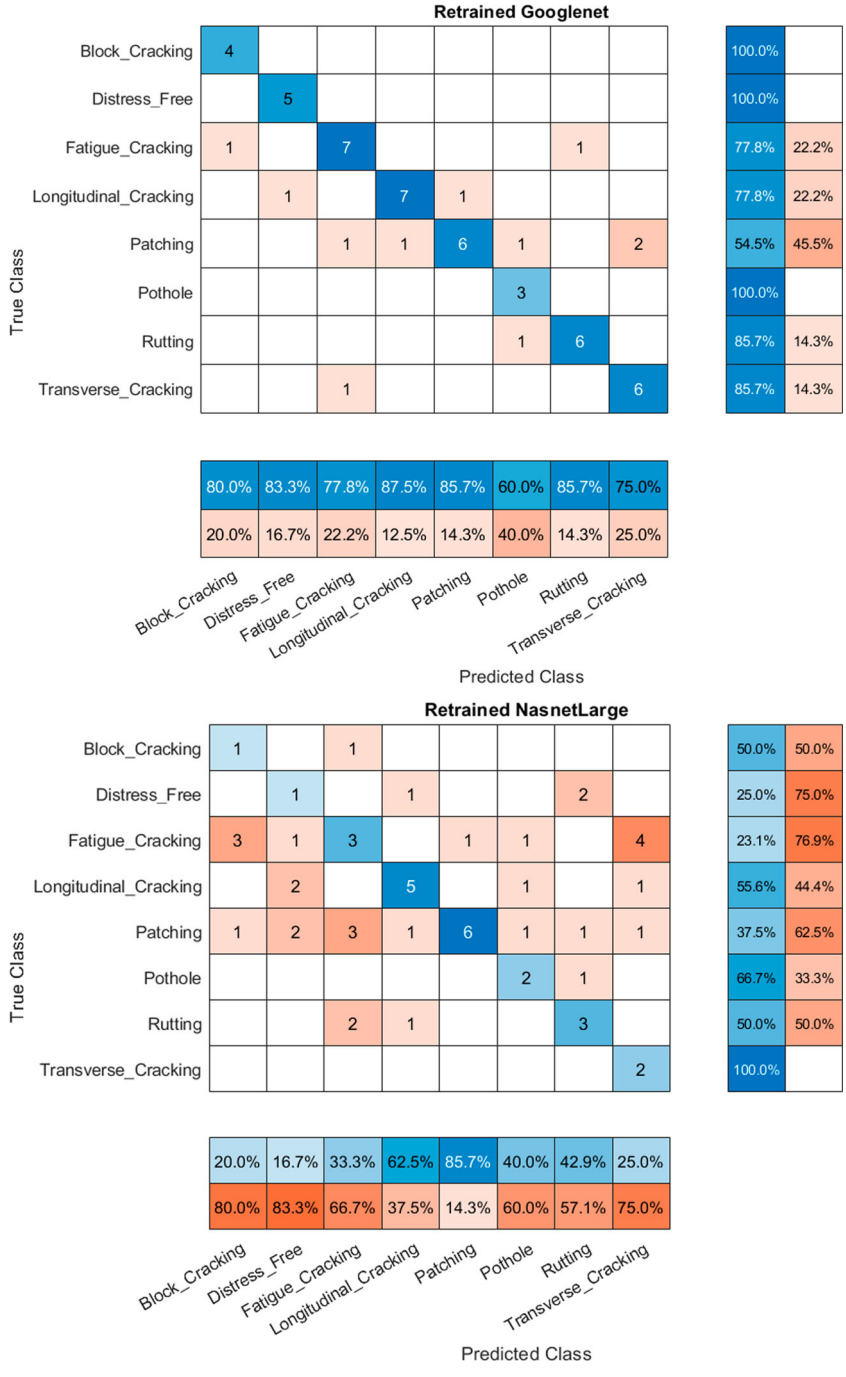

Figure 4. Comparison of network performance based on confusion matrix for selected networks.

rate) and incorrectly classified (false negative rate). The ratio of the sum of all the diagonal values (=44 in the case of Googlenet) divided by the total number of images in the validation set (=55) gives the overall accuracy of the model for the multi-class image classification at hand. Thus, for the results depicted in Figure 4, the overall accuracy values are 80% and 42% for Googlenet and NasnetLarge, respectively. The results from the CM for each of seven networks were used to estimate multiple performance measures including F1-score, precision, recall, specificity, and Matthews correlation coefficient.

The first sub-plot in Figure 4 is the confusion matrix for the Googlenet classifier trained to classify asphalt pavement distresses. In this figure, the first eight diagonal cells show the number of correct classifications by the trained network. It can be seen that seven distresses were classified correctly as longitudinal cracks. This corresponded to 12.7% of all the 55 distresses used for validation of the trained model. Similarly, one of transverse cracks was classified incorrectly as fatigue cracking. This corresponded to approximately 5.5% of all distresses used during validation. Similarly, two of the transverse cracks were classified incorrectly as patching and this corresponded to approximately 3.6% of all validation data. All of the block cracking were correctly predicted as block cracking. Of the fatigue cracking cases, 77.8% were classified correctly, while 22.2% were classified as block cracking and rutting. It is noted that the performance of the trained classifier on patching was the least accurate, as only 54.5% of the patching were classified correctly and 45.5% were classified incorrectly as fatigue cracking, longitudinal cracking and transverse cracking. The overall accuracy of the Googlenet-trained classifier was 80.0% for classifying the eight distresses considered.

The Densenet201 network classified several images correctly (overall accuracy 76%) and was considered the second-most accurate of the seven networks considered, based on the elements of the confusion matrix. The network appeared to have most trouble with longitudinal cracking images, classifying most as distress free, fatigue cracking, or patching.

The trained Squeezenet network classified several images fairly correctly (overall accuracy 66%). The network appeared to have serious difficulty with rutting images (29% accuracy) and patching (43% accuracy). The network misclassified many rutting images as fatigue cracking, longitudinal cracking or patching.

The trained Resnet50 network classified several images of pavement distress correctly (overall accuracy 69%). Unlike Googlenet, the Resnet50 network appeared to have no trouble with pothole images, classifying all of them as potholes. However, the network had trouble classifying block cracking, fatigue cracking, and longitudinal cracking. For instance, the network classified many fatigue cracks as block cracking, longitudinal cracking, patching, pothole, or rutting. The model had a sensitivity (recall) of 71%, which is comparable to previously reported values for TL-based pavement classification tasks in the literature (Ma *et al.* 2017, Maeda *et al.* 2018, Chen *et al.* 2022).

The trained Xception network classified several images of pavement distress correctly (overall accuracy 75.5%). Xception predicted block cracking and distress-free pavements perfectly (100% accuracy). However, the network had trouble classifying all the other distresses including fatigue cracking (78%), longitudinal cracking (50%), patching (43%), potholes (40%), rutting (86%) and transverse cracking (67%). Overall, the network classified many fatigue cracks as block cracking or pothole.

The trained Nasnetlarge network misclassified most of the distress images (overall accuracy 42%). The network appeared to have trouble with all but patching images, classifying many patches as patches and only one as fatigue cracking.

Table 3 shows a summary of the performance measures achieved by each trained classifier based on eight widely used measures for evaluating classifier performance. The data shows Googlenet had the highest F1 accuracy, while Nasnetlarge was ranked the worst. The F1 performances for the middle four networks were comparable. It can be seen from Table 3, Googlenet, Densenet201 and Xception ranked highest while NasnetLarge ranked consistently lower than any other model, irrespective of network.

4.2. Matthews correlation coefficient

Matthews correlation coefficient (MCC) is considered to be a more reliable statistical rate which produces a high score only if the prediction obtained correct results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives), proportional both to the size of positive elements and the size of negative elements in the dataset (Chicco and Jurman 2020). The results are summarised in Table 4.

As shown in Table 4, classifier performance varied depending on the classifier type as well as distress type. It was observed that, while some networks based on Googlenet, Densenet and Xception performed comparatively well across all distress types, others, such as Nasnetlarge, performed poorly across the board. The latter observation was unexpected as Nasnetlarge performed excellently (top 1 accuracy of 82.5%) when used to classify common objects in the

Table 3. Performance measures for seven DCNNs retrained to classify eight pavement distresses.

Measure	Googlenet	Densenet201	Squeezenet	Alexnet	Restnet50	Xception	NasnetLarge
Accuracy	0.800	0.764	0.655	0.636	0.691	0.755	0.418
Precision	0.852	0.795	0.743	0.626	0.699	0.772	0.510
Sensitivity	0.794	0.770	0.653	0.634	0.701	0.757	0.408
Specificity	0.971	0.966	0.949	0.948	0.956	0.965	0.916
F1-score	0.809	0.776	0.671	0.626	0.680	0.739	0.418
MCC	0.594	0.546	0.502	0.487	0.470	0.437	0.312
AUC	0.928	0.988	0.840	0.886	0.884	0.983	0.816
OPT	0.800	0.800	0.600	0.200	0.400	0.800	0.000

Note: OPT = Optimal operating point of the ROC curve.

Table 4. Comparison of performance of various networks ranked by distress type using MCC.

Distress type	Googlenet	Densenet201	Squeezenet	Alexnet	Xception	Resnet50	NasnetLarge
Block cracking	72.3	72.4	71.0	50.0	49.5	30.9	30.0
Distress free	72.0	57.6	57.6	71.6	46.5	42.1	11.1
Fatigue cracking	52.6	45.9	25.9	63.0	46.0	70.1	6.9
Longitudinal cracking	53.4	44.2	56.3	35.0	40.2	40.1	35.0
Patching	36.1	28.9	30.0	37.2	37.2	41.0	24.8
Pothole	69.6	54.8	71.7	39.0	32.1	49.2	42.9
Rutting	59.6	59.8	41.3	37.2	43.2	41.0	30.0
Transverse cracking	59.2	73.4	47.7	56.3	54.4	61.5	69.2

ImageNet (Table 2). The results suggest that it essential to take great care when selecting existing pre-trained networks to classify pavement distress. Another interesting observation from Table 4 was that Densenet201 appeared to be the most versatile classifier as it appeared in the top-three-performing networks for five out of the eight distress categories considered in the study.

As mentioned earlier, because MCC generates a high score only if the classifier was able to predict most of the positive and negative data instances correctly, the measure is often preferred to the F1 score by experts in the field (Akosa 2017). The results of the study suggested that the classifier with the top MCC ranking (Googlenet) performed well on both recall (t_p rate = 0.85) and specificity (t_n rate = 0.97), which agreed with the F1 score ranking. While Squeezenet was ranked third using MCC, Alexnet was ranked third based on the F1 score. In this case, the recall value for Squeezenet was relatively better (t_p rate = 0.74 and t_n rate = 0.95) compared with Alexnet (t_p rate = 0.72 and t_n rate = 0.96). It was observed also that, for the dataset and networks evaluated in this study, image size was strongly correlated with classifier performance(R^2 = 0.75); models with the lower input sizes (e.g. 224 × 224 pixels) consistently outperformed models that required larger input sizes (e.g. 331 × 331 pixels). While no firm conclusions could be made based on the limited number of image sizes used in the current study, the authors believe computer memory could be a factor as the model that performed worse was also the most complex in terms of file size. The findings warrant further investigation to establish the impact of image size and hardware limitations on model performance. All the models performed poorly in the prediction of patching, as the MCC values for each distress type were all below 50%. The fine-grained nature of patching images could have been the reason for universally poor prediction of the distress by the seven networks evaluated. Compared with the F1 score, the ranking of the best performing, and worst performing classifiers appeared to be similar for the distresses and networks used for the study, although MCC appeared to be more sensitive.

As shown in Table 5, average false negative rates for the various classifiers varied from approximately 14.8% for Googlenet to 49.0% for Nasnetlarge. The ranking of classifier performance using the false negative (f_n) measure was similar to that based on MCC. For asphalt pavements, a high false negative rate is undesirable, as high percentages of distresses that might require urgent attention could be missed. The results suggested that multiple performance measures including F1 score, MCC and f_n, could all be used when evaluating individual networks to classify asphalt pavement distresses. On this basis, the results suggested that, for the networks considered in this study, models based on Googlenet and Densenet201 appeared to perform best, while models based on Nasnetlarge performed worst. The results were unexpected because it is commonly assumed that the best performing CNN models in the ILSVRC challenge would also be the top performing models in other visual tasks when using features obtained for the relevant purpose (Kornblith *et al.* 2018). Similar, unexpected results have been reported when models trained on the Imagenet dataset are used in transfer learning to classify images from entirely different domains such as those that were the focus of this study (pavement distress images)

4.3. Multi-metric evaluation

One of the main gaps identified in existing TL-based classification studies on pavement distresses is the lack of information on the use of multiple performance metrics for robustly evaluating network performance, especially metrics that are not sensitive to class imbalance. Class imbalance (e.g. low proportions of potholes in existing datasets in developed countries such as U.S.A. and Japan) has been identified as a major problem, majority of the studies reviewed used individual metrics such as accuracy, precision, recall and F1-score, often in isolation, which might not be reliable when significant class imbalance exist in the dataset. To address this gap, model performance in this study was assessed using a combination F1-score, AUC, model size, and speed of training in a form that will permit easier comparison with existing TL-

Table 5. Comparison of performance of various networks using false negatives. Well-performing networks are characterised by lower false negative rates.

Distress type	Googlenet	Densenet201	Squeezenet	Alexnet	Xception	Resnet50	NasnetLarge
B.cracking	0.0	0.0	0.0	25.0	28.6	50.0	50.0
Distress free	0.0	16.7	16.7	0.0	33.3	37.5	75.0
F.cracking	22.2	30.0	58.8	11.1	28.6	0.0	76.9
L. cracking	22.2	33.3	16.7	44.4	37.5	37.5	44.4
Patching	45.5	50.0	50.0	40.0	40.0	37.5	62.5
Pothole	0.0	20.0	0.0	40.0	50.0	28.6	33.3
Rutting	14.3	14.3	33.3	46.2	33.3	37.5	50.0
T. cracking	14.3	0.0	30.0	16.7	20.0	12.5	0.0

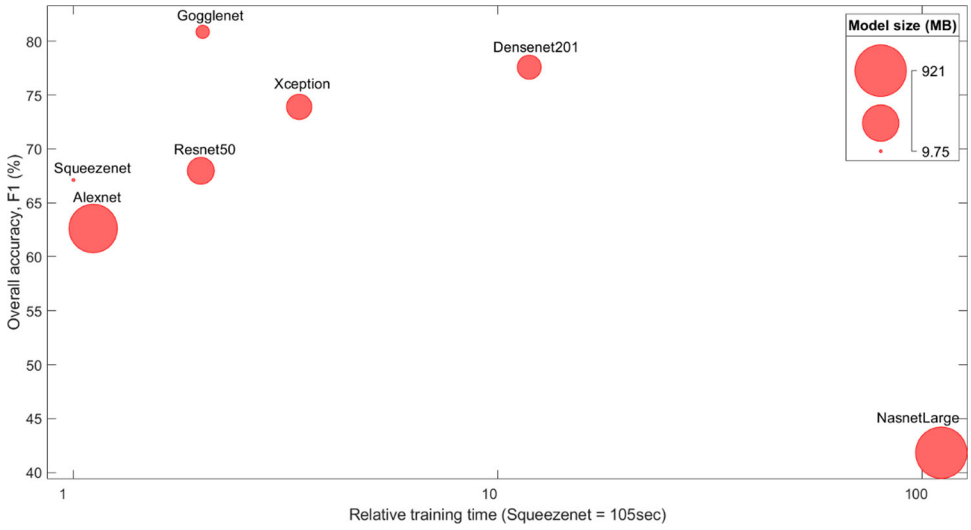

Figure 5. Evaluation of TL-based models using multiple performance metrics. Comparison of the training speeds versus F1-score for seven DCNNs trained to classify asphalt pavement distress.

based models as well as new TL models that will be developed in the future. The results are depicted in Figures 5 and 6, for F1-score and AUC, respectively. From plots and Table 3, the performance levels of the seven models can be clearly identified. The top performing models were characterised by high F1-scores, high AUC, low training times and low model size. For example, Googlenet with F1-score of 80%, also had one of the smallest model sizes as well as training speed that is orders of magnitude smaller than the Nasnetlarge, the lowest ranked model in this study. The top three performing models had similar OPT of 0.80, AUC that ranges between 0.928 and 0.983, and F1-score that lies between 74% and 81%. On the other hand, the lowest performing model has an OPT of 0.0, AUC of 0.816, F1-score of 42%. On the basis of size, the lowest performing model (Nasnetlarge) was almost 14 times as big as the one of the top performing models (Googlenet). The trained network based on Googlenet was almost six times as fast, and three times smaller, than the Densenet201 model. Compared to Table 3, the utility of plots such as those depicted in Figures 5 and 6 is apparent. For example, using the four performance metrics (accuracy, precision, recall and F1-score) commonly used in previous studies, the difficulty of discriminating the best performing model is easy to see. In this case, when considering recall (sensitivity), all the models with the exception of worst performing model, lies in the narrow range of 0.63–0.79, which makes choosing the best performing models difficult. The approach used in this study that is based on multiple performance metrics appears to be more discriminately than commonly used single parameter metrics and could be recommended for future studies.

4.4. Ambiguity of classification

Ambiguity of a classification is defined as the ratio of the second-largest probability to the largest probability, and ranges between zero (nearly certain classification) and 1 (likely

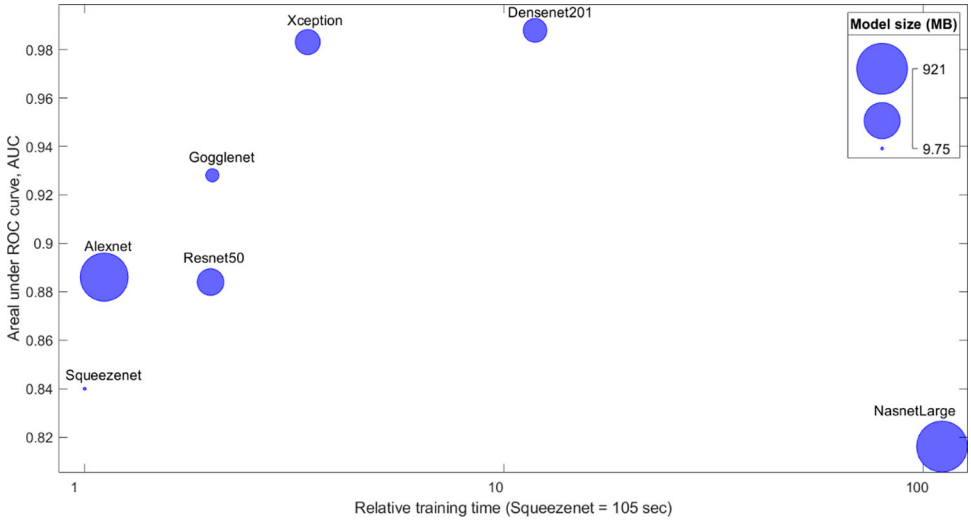

Figure 6. Evaluation of TL-based models using multiple performance metrics. Comparison of the training speeds versus AUC for seven DCNNs trained to classify asphalt pavement distress.

Table 6. Ambiguity parameters for best-performing Googlenet model.

Image #	Ambiguity	Likeliest	Second	True Class
2	0.4899200	Longitudinal Cracking	Fatigue Cracking	Block Cracking
29	0.4265500	Patching	Rutting	Patching
19	0.0161550	Fatigue Cracking	Block Cracking	Fatigue Cracking
14	0.0147900	Fatigue Cracking	Block Cracking	Fatigue Cracking
34	0.0082732	Patching	Longitudinal Cracking	Patching
3	0.0082692	Block Cracking	Longitudinal Cracking	Block Cracking
33	0.0025295	Patching	Rutting	Patching
22	0.0022393	Longitudinal Cracking	Block Cracking	Longitudinal Cracking
6	0.0021190	Distress-free	Longitudinal Cracking	Distress-free
25	0.0018898	Fatigue Cracking	Patching	Longitudinal Cracking
31	0.0017293	Patching	Rutting	Patching
53	0.0015011	Transverse Cracking	Pothole	Transverse Cracking
52	0.0009553	Transverse Cracking	Pothole	Transverse Cracking
27	0.0009103	Longitudinal Cracking	Distress-free	Longitudinal Cracking
51	0.0008291	Transverse Cracking	Rutting	Transverse Cracking
47	0.0007131	Rutting	Distress-free	Rutting
46	0.0005390	Rutting	Fatigue Cracking	Rutting
28	0.0005176	Longitudinal Cracking	Patching	Longitudinal Cracking
10	0.0004625	Distress-free	Longitudinal Cracking	Distress-free
43	0.0003619	Rutting	Longitudinal Cracking	Rutting

to be classified to the most likely class as the second class). Table 6 shows comparison of likeliest prediction with true class for Googlenet. As shown in Table 6, ambiguity ranged from a minimum of 0.00036 to a maximum value of 0.490. In 18 out of 20 observations, the network predicted the true class (shaded green). The misclassification of block cracking as longitudinal cracking or of longitudinal cracking as fatigue cracking is a recognised problem in the field, even for trained technicians. The results demonstrated high precision of the trained Googlenet and agreed with the other performance measures discussed already in this paper. Table 7 shows the corresponding ambiguity of classification results for Nasnetlarge. Ambiguity varied from 0.580 to 0.999, indicating a generally poor predictive performance of the network. Comparing the network prediction of the likeliest class with the true class, as shown in Table 7, the Nasnetlarge trained in this study was able to predict 5 out of 20 images correctly (20%). The results presented in Tables 6 and 7 demonstrated the potential utility of the ambiguity parameter in robustly evaluating the performance of existing deep convolutional neural networks retrained to classify asphalt pavement distresses. For example, results could be used to provide guidance on which models are the best suited for which distresses and also which distresses could cause confusion for which model(s).

4.5. Network evaluation using t-SNE plots

Figure 7 shows a t-SNE plot of the SoftMax activations for Googlenet and Nasnetlarge, the best-performing and worst-performing networks used in this study. The plot shows details of the structure of the posterior probability distribution used by each network for distress classification. The plot shows eight distinct clusters for the Googlenet observations, whereas the Nasnetlarge clusters are not resolved very well. Similar to the confusion matrix-based measures, the ambiguity parameter, and the Matthews correlation coefficient scores, the t-SNE plot suggested that the trained Googlenet network was more accurate at classifying asphalt pavement distresses into eight different classes than Nasnetlarge.

Table 7. Ambiguity parameters for worst-performing model.

Image #	Ambiguity	Likeliest	Second	True Class
50	0.9993900	Transvers Cracking	Longitudinal Cracking	Transverse Cracking
53	0.9627900	Transvers Cracking	Fatigue cracking	Transverse Cracking
48	0.9443100	Transvers Cracking	Longitudinal Cracking	Transverse Cracking
25	0.9179300	Fatigue Cracking	Pothole	Longitudinal Cracking
54	0.9154100	Block Cracking	Fatigue cracking	Transverse Cracking
41	0.8931600	Pothole	Rutting	Rutting
8	0.8560900	Fatigue Cracking	Rutting	Distress-free
32	0.8371800	Fatigue Cracking	Rutting	Patching
36	0.8300100	Longitudinal Cracking	Fatigue cracking	Pothole
28	0.8286500	Longitudinal Cracking	Patching	Longitudinal Cracking
22	0.8152900	Fatigue Cracking	Patching	Longitudinal Cracking
24	0.7909900	Longitudinal Cracking	Block Cracking	Longitudinal Cracking
49	0.7849100	Longitudinal Cracking	Transverse Cracking	Transverse Cracking
51	0.7283200	Fatigue Cracking	Rutting	Transverse Cracking
44	0.6627600	Fatigue Cracking	Rutting	Rutting
30	0.6305300	Rutting	Fatigue cracking	Patching
55	0.6278900	Fatigue Cracking	Pothole	Transverse Cracking
15	0.6172100	Patching	Fatigue cracking	Fatigue Cracking
52	0.5875700	Fatigue Cracking	Pothole	Transverse Cracking
34	0.5796300	Patching	Fatigue cracking	Patching

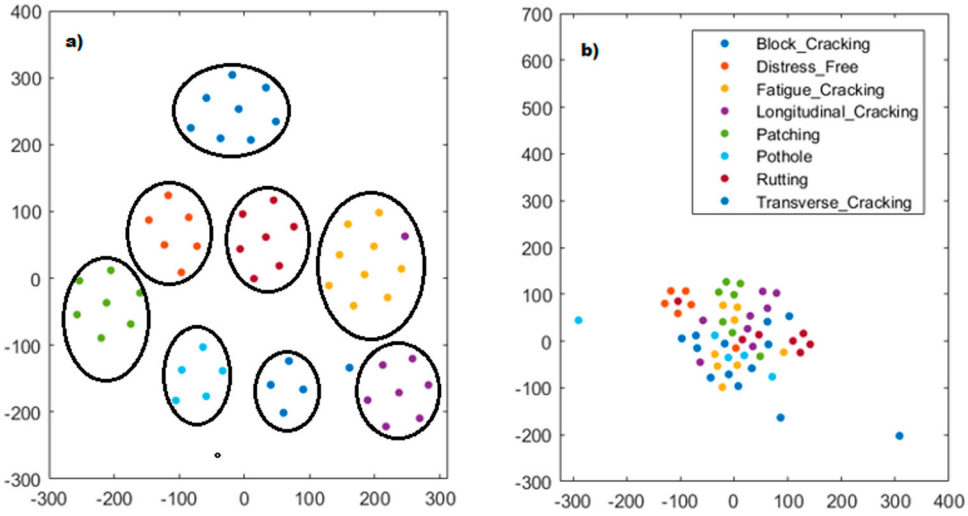

Figure 7. Plot of the SoftMax activations, showing the structure of the posterior probability distribution for Googlenet (a) and Nasnetlarge (b). The legend shows details of the labels for each class

4.6. Discussion

The preliminary results in this study suggest that the specific TL techniques adopted in this study can be applied to Googlenet, Densenet201 or Xception networks to develop fairly accurate and robust pavement distress classification models and warrant further studies. The approach adopted in this study, including the multiple performance measures used, will be of great interest to other pavement engineers considering the adoption of machine learning techniques, such as TL-based DCNNs. The results are by no means universal and it is conceivable that a different combination of TL techniques including hyperparameter optimisations could results in better performance even for pretrained models that performed poorly in this study.

One of the goals of evaluating pavement condition is to understand the mechanisms responsible for causing a given distress in order to recommend remedial action(s) that are likely to result in the most cost-effective solution. Therefore, pavement engineers try to identify the most likely types of distress and their extent, as well as to avoid missing a particular distress that is actually present. The basic ability of a deep neural network to separate pavement distresses into eight classes, as demonstrated in this study, does not facilitate solving such a complex engineering problem directly. Once robust models have been developed or identified, the more complex problem of automatically establishing the condition of the pavement by quantifying the extent of the distress, cause(s) of the distress and suggestion for remedial action need to be developed before a truly automatic system can be achieved. Moreover, the accuracy of the eight-class classification should be regarded with caution since it is affected also by the capability of the annotating technicians to correctly categorise asphalt pavement distresses consistently using photographic images, which has been observed to be not perfect always. However, the evaluations documented in this study are useful in providing insights into the workings of state-of-the-art, deep neural network learning that are necessary to modify these networks to suit pavement applications in the field. The best performing network (Googlenet), when considering a linear combination of F1-score, AUC, OPT, training time, and model size, had a relative score approximately 6 and 8 times better than the next two top-performing modes Densenet201 and Xception, respectively. The results demonstrate the sensitive nature of the multi-metric approach introduced in this study. It should be noted that all the metrics were equally weighted. However, to obtain the linear combination for each network, we mapped each performance metric to a common index range and added then them together.

The work presented in this paper provides an example that shows how TL-based DCNNs techniques can be used in pavement engineering, especially to classify asphalt pavement distress. The pavement engineering field can benefit considerably from the advances made in machine learning, especially in transfer learning. Future work undertaken by the researchers will be focused on many valuable applications and goals, such as predicting distress initiation and propagation and, ultimately, better understanding of the mechanisms that causing a given distress in the first place. The results presented show the potential of using DCNNs to assess pavement distress accurately and eventually, to automate currently tedious and user-dependent tasks to evaluate pavement conditions.

5. Conclusions

In this study, TL techniques were used to retrain seven existing DCNNs to classify approximately 400 images into eight pavement distress class labels. The following conclusions are based on the results obtained using the specific TL techniques implemented in the study:

(1) The results show some existing DCNN's are better than others for developing pavement distress classification models using the specific TL approach adopted in the study. For example, Googlenet, based on the Inception architecture, was found to be the most successful network, with overall accuracy of approximately 80%, while Nasnetlarge, based on the reinforcement learning concept, had

the least accuracy of all the networks considered. The results were unexpected but not surprising, as previous studies have shown that network that are successful in the ImageNet Challenge do not always perform well using transfer-training. The selection of models based only on performance in the classification of ImageNet data (a benchmark in the field) is not recommended. Nasnetlarge has the highest accuracy in ImageNet classification but performed worst when transfer-trained to classify pavement distress in the current study using the specified TL-techniques

(2) Based on a linear combination of F1-score, AUC, OPT, training time, and model size, the best performing network (Googlenet), had a relative score approximately 6 and 8 times better than the next two top-performing modes Densenet201 and Xception, respectively. The results demonstrate the high sensitivity of the multiple metrics approach adopted in this study.

(3) The best-performing networks were characterised by lower proportions of false negative values, very low ambiguity scores, and well-defined t-SNE clusters that showed clear separation between the eight classes of distress considered.

(4) Poor-performing networks were characterised by high proportions of false negatives (i.e. models failed to identify distresses where they actually existed). For pavements, this is undesirable because high percentages of distresses that might require urgent attention could be missed.

(5) Differences were observed in terms of the ability of each model to classify each of the eight pavement distresses considered. It was observed that, while the performance of Nasnetlarge was worst overall in terms of accuracy when all eight distresses were considered, the model performed fairly well in classifying patching.

(6) Even though the predictions of the two top-performing models (Googlenet and Densenet201) were similar both in validation and verification tests, there were some key differences in terms of speed of prediction and model size on file. The trained network based on Googlenet was six times as fast as, and three times smaller than, the Densenet201 model.

(7) It is recommended that future studies are focused on image quality and quantity as a means of improving on the performance of the developed models in terms of prediction speed, accuracy, and model size. Furthermore, the impact of variability in the selection of hyper-parameters on model performance requires further studies. Finally, the results showed that while some TL-networks developed in this study were weak learners overall, they were nonetheless very good as classifier for some individual distresses. Thus, future studies aimed at combining multiple DCNNs to work synergistically to achieve better predictive performance is warranted.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Akosa, JS, 2017. Predictive accuracy: a misleading performance measure for highly imbalanced data. *Proceedings of the SAS Global Forum 2017 Conference*. Cary, North Carolina: SAS Institute Inc., 2017, 942.

Bengio, Y, Courville, A, and Vincent, P, 2013. Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (8), 1798–1828. doi:10.1109/tpami.2013.50. PMID 23787338. S2CID 393948.

Bengio, Y, LeCun, Y, and Hinton, G, 2015. Deep learning. *Nature*, 521 (7553), 436–444. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096.

Bottou, L, 2010. Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT*, 2010, 177–186.

Chen, C, *et al*., 2022. Deep learning-based thermal image analysis for pavement defect detection and classification considering complex pavement conditions. *Remote Sensing*, 14 (1), 106. doi:10.3390/rs14010106.

Chicco, D., and Jurman, G, 2020. The advantages of the Matthews Correlation Coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21, 1.

Chollet, F., 2016. Xception: Deep learning with depthwise separable convolution. Available from: https://arxiv.org/pdf/1610.02357.pdf.

Dean, J., *et al*., 2012. Large scale distributed deep networks. *NIPS 12: Proceedings of the 25th International Conference on Neural Information Processing Systems, Volume 1*, December 2012, 1223–1231.

Deng, J, *et al*., 2009. ImageNet: a large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. doi: 10.1109/CVPR.2009.5206848.

Duntsch, I., and Gediga, J, 2019. 3rd international Conference on Machine Vision and Information Technology (CMVIT 2019). *Journal of Physics: Conference Series*, 1229, 011001.

Gopalakrishnan, K., *et al*., 2017. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 157, 322–330.

Gopalakrishnan, K., *et al*., 2018. Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model. *International Journal of Traffic and Transportation Engineering*, 8 (1), 1–14.

Gopalakrishnan, K, 2018. Deep learning in data-driven pavement image analysis and automated distress detection: a review. *Data*, 3 (3), 28. doi:10.3390/data3030028.

He, K., *et al*., 2016. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. doi: 10.1109/CVPR.2016.90.

Huang, G, Liu, Z, and van der Maaten, L, 2018. Densely connected convolutional networks. arXiv:1608.06993.

Iandola, FN, *et al*., 2016. Squeenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 Mb model size. arXiv: *Computer Vision and Pattern Recognition*. https://arxiv.org/pdf/1602.07360.pdf.

Kornblith, S, Shlens, J, and Le, QV, 2018. Do better ImageNet models transfer better? http://arxiv.org/abs/1805.08974.

Krizhevsky, A., Sutskever, I., and Hinton, G. E, 2012. Imagenet classification with deep convolutional neural networks. *Proc., 25th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada*, 1097–1105.

Ma, K., Hoai, M., and Samaras, D, 2017. Large-scale continual road inspection: visual infrastructure assessment in the wild. *Proceedings of the British Machine Vision Conference, London, UK*, 4–7.

Maeda, H., *et al*., 2018. Road damage detection using deep neural networks with images captured through a smartphone. doi:10.1111/mice.12387.

Majidifard, H., *et al*., 2020. Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses. *Transportation Research Record: Journal of the Transportation Research Board*, 2674 (2), 328–339.

Mandal, V., Uong, L., and Adu-Gyamfi, Y., 2018. Automated road crack detection using deep convolutional neural networks. *2018 IEEE International Conference on Big Data (Big Data)*.

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

O'Mahony, N, 2020. Deep learning vs. traditional computer vision. *In*: K. Arai and S. Kapoor, eds. *Advances in Computer Vision. CVC 2019. Advances in Intelligent Systems and Computing*, vol. 943. Cham: Springer. doi:10.1007/978-3-030-17795-9_10.

Pan, S.J., and Yang, Q, 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22, 1345–1359.

Peraka, NSP, Biligiri, KP, and Kalidindi, SN, 2021. Development of a multi-distress detection system for asphalt pavements: transfer learning-based approach. *Transportation Research Record*, 10, 538–553. doi:10.1177/03611981211012001.

Ranjbar, S., Nejad, F.M., and Zakeri, H, 2021. An image-based system for pavement crack evaluation using transfer learning and wavelet transform. *International Journal of Pavement and Research Technology*, 14, 437–449. doi:10.1007/s42947-020-0098-9.

Schmidhuber, J, 2015. Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.

Shin, H. C., *et al*., 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35, 1285–1298.

Simonyan, K., and Zisserman, A, 2015. Very deep convolutional neural networks for large-scale image recognition. *International Conference on Learning Representations*, 1–14.

Siriborvornratanakul, T, 2018. An automatic road distress visual inspection system using an onboard in-car camera. *Advances in Multimedia*, 2018, 1–10.

Szegedy, C., *et al*., 2015a. Going deeper with convolutions. *Computer Vision and Pattern Recognition*, 1–9. doi:10.1109/CVPR.2015.7298594.

Szegedy, C., *et al*., 2015b. Rethinking the inception architecture for computer vision. arXiv:1512.00567. doi:10.1109/CVPR.2016.308.

Szegedy, C., Ioffe, S., and Vanhoucke, V, 2017. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: *AAAI*, 4278–4284.

van der Maaten, L., and Hinton, G., 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.

Vavrik, W., *et al*., 2013. *PCR evaluation – Consider transition from manual to semi-automated pavement distress collection and analysis*. Ohio Department of Transportation. Office of Statewide Planning and Research.

Wu, Z, *et al*., 2016. Deep Learning for Video Classification and Captioning. *In*: Chang Shih-Fu, ed. *Frontiers of Multimedia Research*. New York: Association for Computing Machinery and Morgan & Claypool.

Zeiler, M., and Fergus, R, 2013. Visualizing and understanding convolutional networks. *European Conference on Computer Vision (ECCV)*, 8689, 818–833.

Zhu, J, *et al*., 2022. Pavement distress detection using convolutional neural networks with images captured via UAV. *Automation in Construction*, 133. doi:10.1016/j.autcon.2021.103991.

Zoph, B., *et al*., 2018. Learning transferable architectures for scalable image recognition. arXiv:1707.07012.

Zoph, B., and Le, Q. V, 2017. Neural architecture search with reinforcement learning. *International Conference on Learning Representations*.