Journal of Civil Structural Health Monitoring (2020) 10:757–773
https://doi.org/10.1007/s13349-020-00409-0

ORIGINAL PAPER

MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure

Vedhus Hoskere[1] · Yasutaka Narazaki[1] · Tu A. Hoang[1] · B. F. Spencer Jr.[1]

Received: 19 November 2019 / Accepted: 20 May 2020 / Published online: 8 June 2020

Abstract

Manual visual inspection is the most common means of assessing the condition of civil infrastructure in the United States, but can be exceedingly laborious, time-consuming, and dangerous. Research has focused on automating parts of the inspection process using unmanned aerial vehicles for image acquisition, followed by deep learning techniques for damage identification. Existing deep learning methods and datasets for inspections have typically been developed for a single damage type. However, most guidelines for inspections require the identification of multiple damage types and describe evaluating the significance of the damage based on the associated material type. Thus, the identification of material type is important in understanding the meaning of the identified damage. Training separate networks for the tasks of material and damage identification fails to incorporate this intrinsic interdependence between them. We hypothesize that a network that incorporates such interdependence directly will have a better accuracy in material and damage identification. To this end, a deep neural network, termed the material-and-damage-network (MaDnet), is proposed to simultaneously identify material type (concrete, steel, asphalt), as well as fine (cracks, exposed rebar) and coarse (spalling, corrosion) structural damage. In this approach, semantic segmentation (i.e., assignment of each pixel in the image with a material and damage label) is employed, where the interdependence between material and damage is incorporated through shared filters learned through multi-objective optimization. A new dataset with pixel-level labels identifying the material and damage type is developed and made available to the research community. Finally, the dataset is used to evaluate MaDnet and demonstrate the improvement in pixel accuracy over employing independent networks.

Keywords Damage detection · Computer vision · Multi-task learning · Semantic segmentation · Structural inspections

1 Introduction

Condition monitoring is an essential step in ensuring the safety and serviceability of civil infrastructure. Detailed information about the current state of a structure provides valuable insights that can be used for a number of applications ranging from prioritization of repairs to the review of design and construction procedures. In the United States, current practice for assessing structural health is predominantly reliant on manual visual inspections [1]. High-profile catastrophic accidents like the I-35W bridge collapse in Minneapolis underscore the fact that manual inspections may miss important details, despite following best practices [2]. For some applications, structural inspections pose unique human challenges. For example, describing the nature of work in a post-disaster scenario, the ATC-20 field manual [3] states that post-earthquake safety evaluations of buildings is "grueling work," resulting in high level of stress on the volunteer inspectors that may lead "burn-out." As another example, inspections of large structures like dams and bridges bring a high level of difficulty as engineers often have to rappel down a surface to inspect for any damage over large areas [4] (see Fig. 1). Thus, the laborious, time-consuming, unsafe, and subjective nature of manual

✉ Vedhus Hoskere
hoskere2@illinois.edu

Yasutaka Narazaki
narazak2@illinois.edu

Tu A. Hoang
tuhoang2@illinois.edu

B. F. Spencer Jr.
bfs@illinois.edu

[1] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61820, USA

Fig. 1 Inspection of the Mike O' Callaghan–Pat Tillman bridge for deficiencies at the Hoover dam

inspections motivate research into methods for automating such inspections.

A natural step forward is to automate parts of the manual inspection process. Self-navigating unmanned aerial vehicles (UAVs) are ideal candidates for data acquisition, but intelligent and automated processing of the large amount of data collected is needed to fully realize the potential of such a system. Among other things, such a system must be adept at identifying different types of damage that may occur in the structure being monitored.

Researchers have developed specific techniques to identify a variety of damage types, e.g., cracks, spalling, and corrosion in structural members made of concrete, steel, and asphalt. Early work focused mainly on identifying concrete cracks by image processing techniques (e.g., [5–9]) working in principle by applying a threshold to the output of hand-crafted image filters. More recently, Paal et al. [10] proposed a combination of segmentation, template matching for spall detection, and assessment on concrete columns. A novel orthogonal transformation combined with a Bridge Condition Index was used in Adhikari et al. [11] to quantify degradation and subsequently map to condition ratings. Chen et al. [12] employed a support vector machine to identify rust on steel bridges, where Bonnin-Pascual and Ortiz [13] used AdaBoost to detect corrosion navigational vessels. Research about fatigue crack detection in civil infrastructure has been fairly limited. Yeum and Dyke [14] manually induced damage on a steel beam to give the appearance of fatigue cracks. Prior work has also been done on crack detection in asphalt pavements. Hu et al. [15] used a local binary pattern (LBP) algorithm to identify cracks. Zhang et al. [16] used filter-based features together with a classifier to identify cracks for subway tunnel safety. To identify the presence of damage, the studies and techniques described thus far typically made certain assumptions on the input and then developed appropriate hand-crafted filters to extract features that were treated with a threshold or a trained classifier. However, the application of such techniques in an automated structural inspection environment is limited, because they rarely employ the contextual information that is available in the regions where damage is present. Developing effective general algorithms is difficult, because real-world situations vary extensively.

The generality of convolutional neural networks (CNNs) and deep learning algorithms (DLAs) in computer vision [17–19] has prompted their use for a wide range of applications in civil engineering including, for example, topology optimization [20–22], structural health monitoring and inspections [23–28], city-scale risk and condition assessment [29–31], construction activity and progress monitoring [32–35], and germane to this article, applications in vision-based damage identification [36–45]. Yeum et al. [36] utilized a CNN for the extraction of important regions

of interest in highway truss structures to ease the inspection process. Narazaki et al. [37–39] focused on reducing the false positive rate by identifying important structural components as a precursor to damage identification. Zhang et al. [40] employed CNNs for the application of crack detection in asphalt pavements. Yeum et al. [41] tested the use of RCNN with spall detection but with limited accuracy (59.39%). Researchers have begun studying multi-damage identification using deep learning employing either object detection or semantic segmentation methods. Object detection involves drawing bounding boxes around the object of interest (e.g., cracks), whereas semantic segmentation involves associating each pixel in the image with a damage class. Hoskere et al. [42] proposed deep learning-based semantic segmentation of damage for six damage types, namely, concrete cracks, spalling, exposed rebar, asphalt cracks, corrosion, and fatigue cracks. Cha et al. [43] used region-based deep learning for object detection of concrete cracks, corrosion, and steel delamination. Rubio et al. [44] studied the identification of multiple types of damage in concrete including delamination and rebar exposure. As the shape of damage observed on civil infrastructure is typically amorphous, the authors of the current work are of the opinion that semantic segmentation is a more suitable approach for multiple damage identification as opposed to object detection. For a comprehensive review of recent advances in vision-based inspections, readers are directed to Spencer et al. [45].

While deep learning methods have proven to be successful for multiple damage identification strategies, from the standpoint of automating inspections, evaluating the significance of the damage also depends on the material type on which they occur [3, 46]. For example, during a bridge inspection, cracks on steel girders under a deck have different implications to those on the concrete or asphalt portions of the deck. Similarly, spalling associated with concrete and asphalt surfaces has different implications. Thus, the identification of material types is an important problem for the ultimate goal of structural inspections.

The naive way to conduct the tasks of material identification along with damage identification would be to train separate deep networks. However, given the highly complementary nature of these tasks, independent networks will likely learn similar features resulting in an inefficient use of computation and memory resources. Instead, an efficient method to conduct these tasks would be to use multi-task learning. Multi-task learning is the process of learning multiple tasks with a single model. Contrary to multi-class segmentation for images where each pixel is assigned one label out of all classes, multi-task learning results in each pixel being assigned as many labels as there are tasks, with each task being a different multi-class segmentation (e.g., damage task: no damage, crack, exposed rebar, etc.; material task: steel, concrete, other, etc.). In a multi-task learning model, a shared representation is used for prediction of all tasks. In addition to providing efficiency, the main benefit of multi-task models is that they help to prevent overfitting and improve network performance [47]. Due to these benefits, multi-task learning has successfully been used for applications like simultaneous depth and semantic segmentation [48], reasoning for autonomous cars [49], detecting multiple modalities in medical imaging [50], and for facial expression understanding [51].

In this paper, we hypothesize that training a single network for the tasks of material and damage identification will allow for improved performance over the naïve case of training separate networks. We propose MaDnet (material and damage network), a multi-task network for the simultaneous semantic segmentation of multiple types of materials (concrete, steel, asphalt), fine damage (cracks, rebar), and coarse damage (spalls, corrosion). The damage task is broken up into two tasks so as to allow for different upsampling filters to be learned for each of the tasks. To learn the features of the network, two different types of combined loss functions are empirically compared. We develop a new dataset with pixel-level material and damage labels, and make it publicly available to the research community. The dataset is used to evaluate the accuracy of MaDnet, compared to the naïve case of training independent networks as in Hoskere et al. [26] for material and damage identification. The results are used to demonstrate the efficacy of sharing features across these tasks. Section 2 of this paper outlines the technical details of the proposed network, Sect. 3 presents the details of a new dataset for multiple structural damage types, and Sect. 4 presents results from experiments. Section 5 presents a discussion of the experimental results. Section 6 presents the main conclusions of the study, followed by the references.

2 Proposed network for material and damage segmentation

This section outlines the proposed network architecture for multi-task semantic segmentation of material and damage types. A fully convolution network (FCN) is used as the basis for development of the proposed network. A schematic of the overall architecture is shown in Fig. 2 and discussed in remainder of this section. The multi-task network architecture schematic shown highlights three main parts of the proposed network, namely, the encoder, the branched decoders, and the combined loss function. The input image is passed through a single encoder to obtain relevant features. These features are fed into three different decoder branches, each of which are trained to

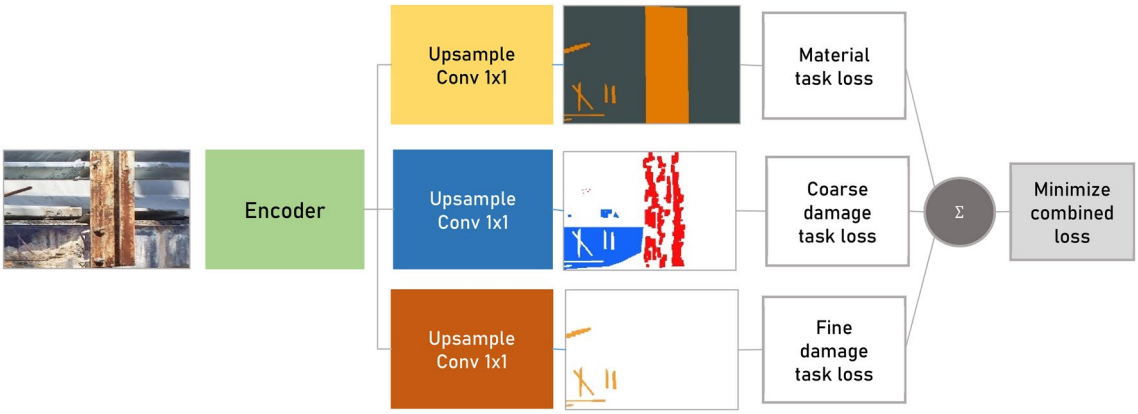

Fig. 2 Schematic diagram of the proposed network

output a different task. The loss functions from each of these tasks are combined using a multi-task loss function that is minimized.

2.1 Fully convolutional networks

FCNs are neural networks with each layer represented in the general form of a convolution [52] as:

$$y_{ij} = f_{ks}\left(\left\{x_{si+\delta i,sj+\delta j}\right\}_{\delta j,\delta i\in 0,k}\right), \tag{1}$$

where y_{ij} represents the output of any layer, x_{ij} represents the input, k is the kernel size, s is the sampling factor, and f represents the layer type, which could be matrix multiplication, spatial max, or an elementwise non-linearity. FCNs can be thought of as a type of CNN with fully connected layers replaced by 1×1 convolution layers. Each pixel in the input image is mapped to a label, resulting in an output label map that is the same size as the input image.

To increase the fineness of the segmentation, 'skip' layers are employed to fuse information learned in earlier feature layers. Starting from the smallest scale, the feature maps are upsampled by a factor of 2, treated with 1×1 convolutions to make predictions, and then summed with the 1×1 convolution predictions from the last feature map at the previous scale. This process of generating these upsampled features is repeated until the output is the size of the input image. The output is treated with a softmax activation function, and the class with the highest probability is predicted as the assigned label for each pixel. A schematic illustration of the network is shown in Fig. 3. The next section describes the specific layers used in the proposed network.

2.2 Single-task network architecture

Each of the single-task networks is trained using a ResNet [17] architecture with 45 layers. The details of the encoder

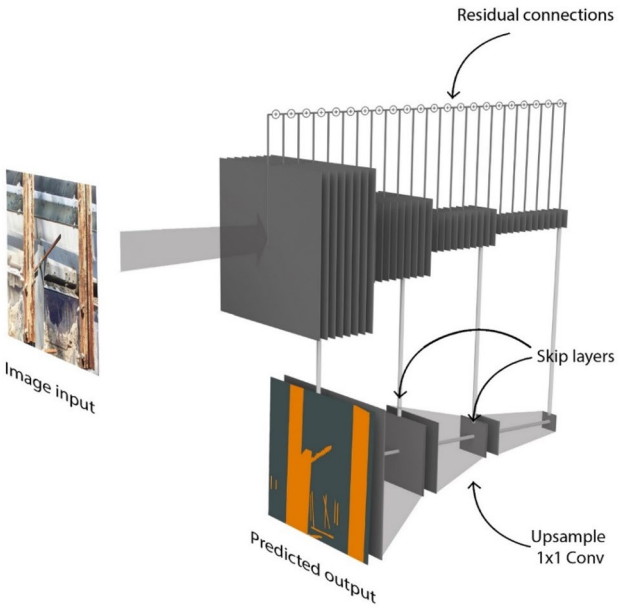

Fig. 3 Schematic illustration of feature layers in the proposed FCN

part of the architecture are provided in Table 1. Residual connections involve the summation of the output of prior layers to enforce learning of new information in subsequent layers. These residual connections are used between alternate layers (e.g., Conv0 to Conv2, Conv2 to Conv4, etc.). A rectified linear unit is used as the non-linearity for all layers of the network. The details of the decoder part of the architecture are provided in Table 2. The skip connections with 1×1 convolutions described in the previous subsection are taken after the Conv8, Conv20, and Conv32 layers.

2.3 Multi-task learning

The principle goal of multi-task learning is to improve the performance on unseen data from the same distribution as

Table 1 Details about the encoder

Layer name (s)	Size	Layer name (s)	Size
Conv0	7×7×64 (stride 2)	Maxpool1	2×2
Conv1–Conv8	3×3×64 (stride 1)	Conv21-Conv32	3×3×128 (stride 1)
Maxpool0	2×2	Maxpool2	2×2
Conv9–Conv20	3×3×64 (stride 1)	Conv33-Conv44	3×3×128 (stride 1)

Table 2 Details about the decoder

Layer name	Encoder connection	Convolution	Upsample
Decoder0	Conv44	–	Bilinear
Decoder1	Conv32	1×1	Bilinear
Decoder2	Conv20	1×1	Bilinear
Decoder3	Conv8	1×1	Bilinear

the training data by learning from multiple objectives or tasks. In other words, the goal is to improve the generality of a model. In a multi-task learning system, Caruana et al. [47] showed that for related tasks, the generalization is improved using the task-specific knowledge contained in different but complementary tasks. Intuitively, additional tasks provide additional perspectives for the network to consider before making predictions. Any prediction on unseen data requires the network to make assumptions (often referred to as "inductive bias" in the machine learning literature) based on representations learned from the training data. When learning multiple complementary tasks, these assumptions are less likely to incorporate task-specific artifacts in the data that cause overfitting and more likely to learn a more general representation resulting in improved performance.

2.4 Multi-task network architecture

The multi-task network is developed using the single-task network described in Sect. 2.2 as the basis. The proposed multi-task network shown in Fig. 4 is based on the work by Kendall et al. [48]. The sharing of the encoder features amongst different tasks helps to regularize the learned features and prevent overfitting. In the developed network, the damage task is divided into two tasks so as to allow the network to learn different upsampling filters for each of the tasks. For example, the edges of fine damage like cracks and exposed rebar are likely to be close to each other and there is benefit in learning specific filters to accomplish this. Thus, there are three tasks in total, two damage tasks and one material task. The encoder of the multi-task network

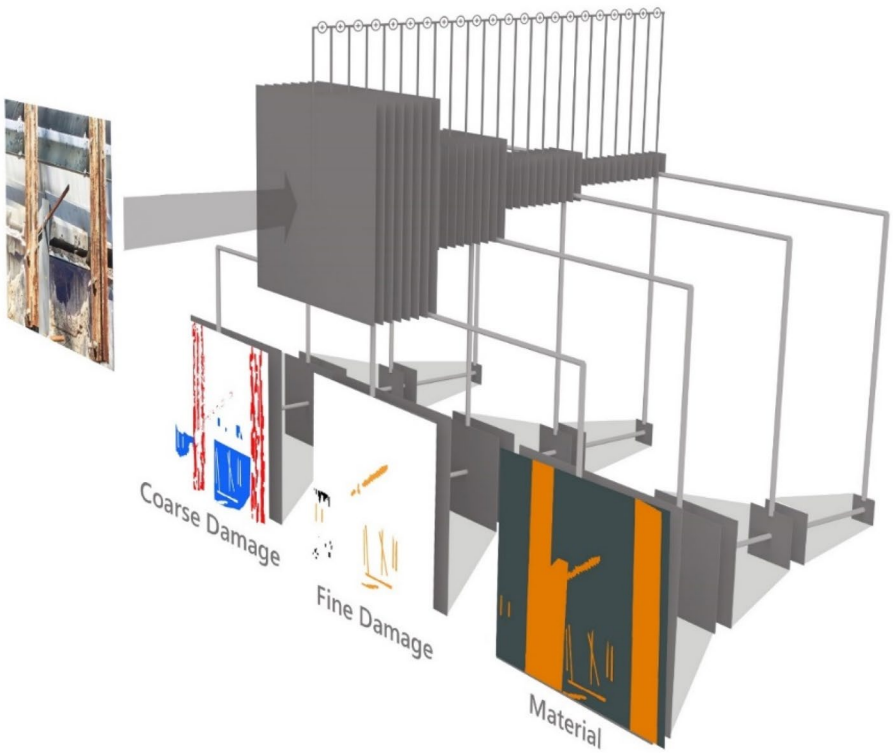

Fig. 4 Schematic illustration of MaDnet

architecture is identical to the encoder architecture defined in Table 1. In the decoder architecture for the multi-task network, three of the decoders described in Table 2 are employed. The decoder layers are built from encoder connections just before the encoder pooling layers with bilinear upsampling followed by 1×1 convolutions. The output of each of these decoders is treated with a softmax function to produce individual loss functions for each of the tasks. Finally, the loss functions are combined using multi-objective loss functions described in the next section.

2.5 Multi-objective loss functions

When combining the individual loss functions from different tasks, a critical decision to be made is determining the weights attributed to individual tasks in the overall loss. Two different multi-task objective functions are examined empirically, namely, a simple additive loss and a homoscedastic loss function. The additive-loss function shown in Eq. (2) is chosen for its simplicity as it allows all tasks to be weighted equally, thus learning features that are relevant to all tasks:

$$L = L_{\mathrm{m}} + L_{\mathrm{cd}} + L_{\mathrm{fd}}. \tag{2}$$

The terms L_{xx} represent the individual loss values. The subscript m denotes material, cd denotes coarse damage, and fd denotes fine damage. Apart from the additive-loss function, a homoscedastic loss function [48] shown in Eq. (3) is also examined to allow the network to learn the weights for each of the tasks directly. Each term in the loss function is weighted by the inverse of a variance that is to be learned during the training process. By tuning the weights, the network tends to provide more importance to tasks that are poorly performing. The homoscedastic loss function thus offers one possible way to obviate assigning weights manually:

$$L = \frac{L_{\mathrm{m}}}{2\sigma_{\mathrm{m}}^2} + \frac{L_{\mathrm{cd}}}{2\sigma_{\mathrm{cd}}^2} + \frac{L_{\mathrm{fd}}}{2\sigma_{\mathrm{fd}}^2} + \log\left(\sigma_{\mathrm{m}}\sigma_{\mathrm{cd}}\sigma_{\mathrm{fd}}\right). \tag{3}$$

The terms σ_{xx} are learned weight factors that determine the importance of each of the loss values in the overall loss function.

2.6 Training considerations

Several techniques commonly used for the training of deep networks were implemented to allow the networks to adequately generalize. The network parameters W and b were trained by minimizing the cross-entropy loss function between the predicted softmax probabilities and the corresponding one-hot labels with an $L2$-regularization weight decay [53]. The incorporation of the weight decay term gives preference to smaller weights and helps tackle overfitting. Batch normalization was applied to address the covariate shift that occurs during training [24], where each feature dimension is shifted by a weighted mean and standard deviation that was learned during training. The percentage of pixels in each of the classes varies significantly. For example, some classes such as cracks have much fewer pixels than spalling or corrosion due to the nature of damage. To balance the frequencies of different classes in the dataset and prioritize all classes equally, median class balancing [26] was applied by reweighting each class in the cross-entropy loss. Data augmentation by resizing and cropping was incorporated to increase the efficacy and efficiency of training and prevent issues such as overfitting. The training was conducted using the Adam optimizer [54] implemented in Tensorflow [55]. To train and evaluate the proposed network, a new labeled dataset with multiple material and damage types was created, as described in the next section.

3 MaDnet dataset

While supervised learning techniques have been highly successful for semantic segmentation [52, 56], the development of trained models requires labeled datasets of a large number of images. In this work, a new dataset is created for the purpose of material and damage segmentation by extending the dataset used in [57]. This section provides details about the developed dataset made available at https://sites.google.com/view/illinois-madnet/home.

3.1 Image details

Different damaged specimens were photographed by the authors, and images of full structures available in the public domain on the Internet were included to construct the dataset. Some online sources of images include: datacentrehub.org [58], bridgehunter.com [59], images available on the websites of the US Army Corps of Engineers [60], as well as images acquired from google image searches. Overall, the assembled dataset includes images of reinforced concrete buildings, steel bridges, concrete bridges, asphalt pavements, hydraulic structures, inland navigation infrastructure, concrete pavements, and damage laboratory specimens. The criteria for selecting images included (1) the presence of visible damage, and (2) representation from a wide variety of structures. The dataset includes a total of 339 images of over 250 different structures of varying sizes that were then divided into 1695 images of a uniform size 600×600.

The number of images required to successfully train CNNs for classification problems is typically very large ($> 100{,}000$) depending on the number of parameters in the network and the network architecture. For semantic segmentation problems, however, research has shown that

a far fewer number of images will suffice, because each pixel serves as a data point. For example, two of the datasets used in Farabet et al. [61] for semantic segmentation were of comparable size to the MaDnet dataset, namely, the Stanford background dataset with 715 images, and the SIFT flow dataset with 2688 images. Other studies have also used semantic segmentation datasets of similar size, for example [62, 63]. The trade-off is that every pixel requires a label and, thus, annotation of each image requires much more time as opposed to the annotation for an image for a classification dataset.

3.2 Image labeling

The images were labeled manually by the authors. A Matlab GUI was created to facilitate the labeling of the images. A screenshot of the GUI is shown in Fig. 5. While several labeling softwares are available online in the public domain, none of them provided the necessary features for semantic segmentation as demanded by this study. The created GUI allows the user to paint over the images to delineate the location of the damage or material type. Different brush colors are available to select different damage/material types and the size of the brush can be changed based on the fineness required. Morphological filters were applied with a manually changeable threshold to help easily select pixels that may correspond to cracks. The masks created by the use of these methods can be further refined by the user depending on the requirements of the image. The next subsection describes the dataset developed using this GUI in more detail.

3.3 Damage and material classes

The dataset was created with images containing one or more of the damage and material classes including, (1) cracks and exposed rebar, (2) spalling and corrosion, and (3) concrete, steel, asphalt, and other material. In the survey of works on using computer vision for identification of damage in civil infrastructure [45], these types of damage were found to be the ones that researchers were most interested in identifying and were thus chosen as the classes for the dataset. The distribution of the images across the different classes in the dataset is provided in Table 3. Some sample images and their corresponding labels are shown in Fig. 6. The color key for the labels is provided at the bottom of the figure.

Table 3 Details of MaDnet dataset of damaged structures created for this study

Material type	Number of images	Damage type	Number of images
Concrete	665	Spalling, exposed rebar	324
		Cracks	341
Steel	595	Corrosion	379
		Cracks	216
Asphalt	435	Cracks	435
Total	1695		

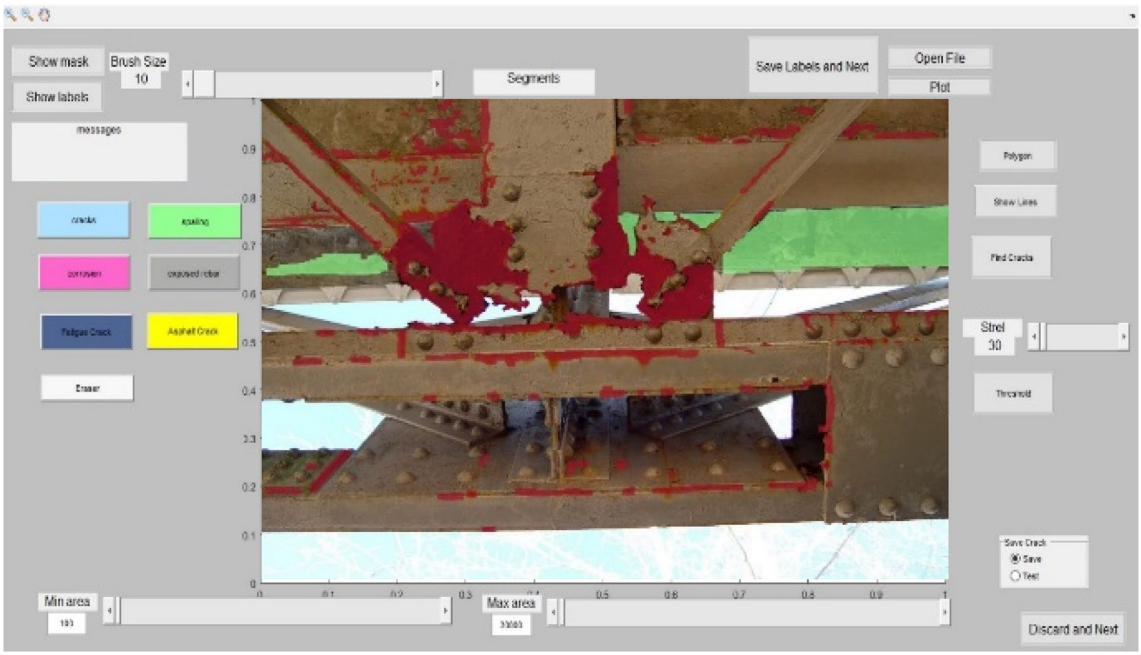

Fig. 5 Graphical user interface for labeling damage in images for semantic segmentation

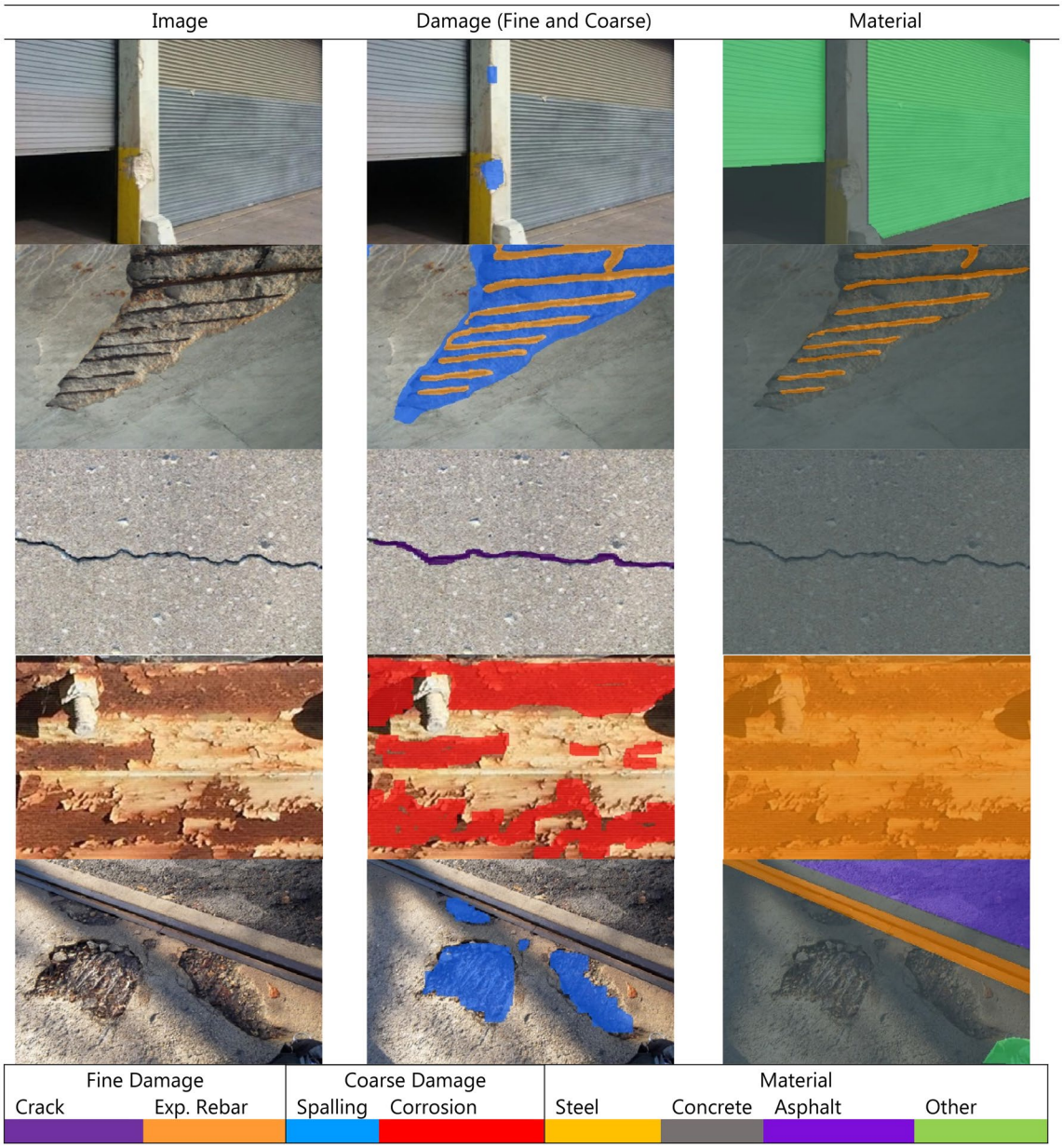

Fig. 6 Sample images from the MaDnet dataset

3.4 Damage scales

Damage in images collected for inspections may occur at multiple scales and, thus, a network should be able to generalize across scales. Figure 7 shows the distribution of area of pixels representing damage for three of the damage classes across all the images to provide an indication of the size of damage in the dataset. All the three histograms follow a long-tailed distribution. The reader may note that the area of pixels for spalling is much higher than cracks and exposed rebar as dictated by the nature of the damage types. The plots indicate a large variation in the sizes of damage in the developed dataset. For example, there are over 60 images with concrete crack area from 0 to 1%, from 3 to 4%. and above 6%. Similarly, with regards to spalling, there are a number of images where only a small portion of the image is occupied by spalling damage as well as some close-up images.

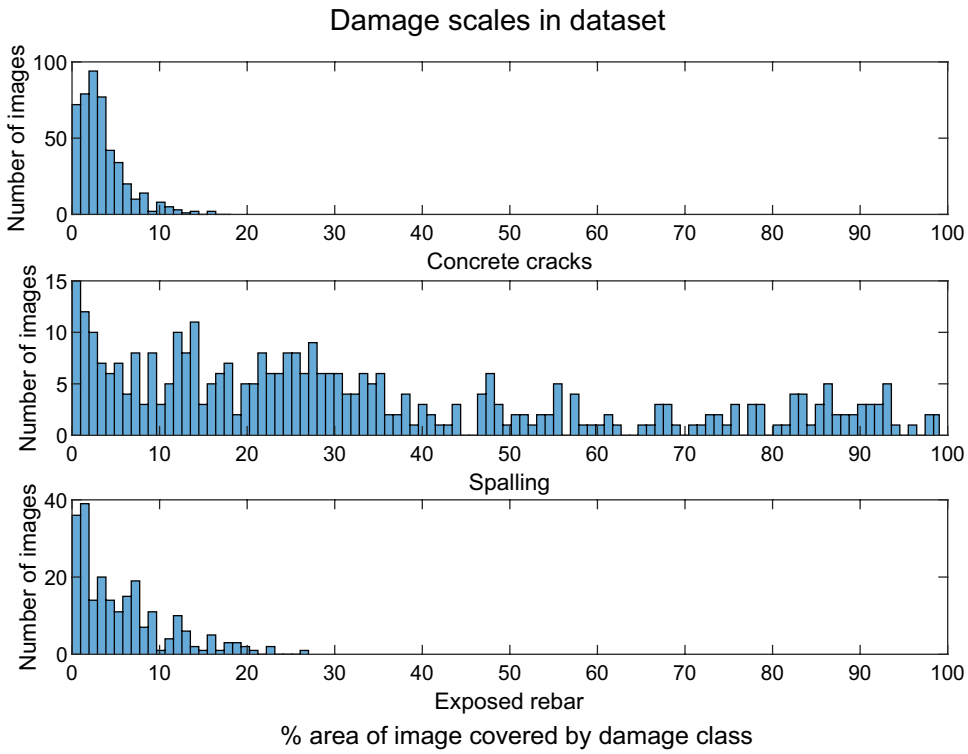

Fig. 7 Damage scales in dataset

4 Experiments and results

The network algorithms described in Sect. 2 were trained on the MaDnet dataset and the proposed multi-task networks with both loss functions were compared to individual single-task networks. This section describes the training process along with results and discussions from different experiments that were conducted.

4.1 Network training

The networks were implemented in Tensorflow [55] and trained from scratch. To address issues with limited graphics memory, the images were resized from $600 \times 600 \times 3$ to $288 \times 288 \times 3$. An online data augmentation strategy was implemented to artificially increase the amount of data based on suggestions made in [61]. Specifically, random resizing was conducted with factors uniformly distributed between 0.75 and 1.25, together with random rotations between $\pm$ 15° and, random flipping and white noise with standard deviation of 2. 70% of the images were used for training purposes, 10% were used for model validation, and the remaining 20% were set aside for testing purposes. For each of the networks, the labeled data were fed in batches of 4 images at a time. The training was carried out on a Windows PC with an i7 7700 2.8 GHz processor, NVIDIA GTX 1070 8 Gb graphics card, and 16 GB RAM.

Table 4 Network training hyperparameters

Training hyperparameters	Value
Learning rate	1e−3
Number of epochs	4000
Train/test/validation split	0.7/0.2/0.1
Data augmentation scale	0.75–1.25
Data augmentation rotation	± 15°
Data augmentation noise	White noise with standard deviation of 2%

The learning rate used for both the networks was 10^{-3}. The training hyperparameters are listed in Table 4.

For a fair comparison between the networks, all networks were trained for the same number of epochs (i.e., cycles through the entire training set). Training was continued until the validation accuracy was seen to significantly deviate from the training accuracy from in any one of the networks. This deviation was first observed in the single coarse damage network, as shown in Fig. 8. This deviation occurred roughly after about 4000 epochs. The training for all the networks was thus stopped after 4000 epochs.

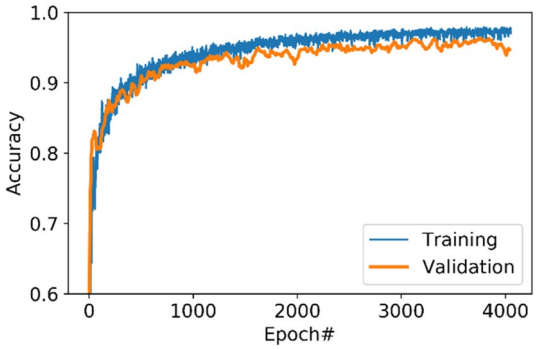

Fig. 8 Training and validation accuracy for single coarse damage network

4.2 Accuracy comparison for different networks

The results from training the multi-task networks are evaluated and compared against the results from single networks with two different metrics: pixel accuracy and the intersection-over-union (IoU). The pixel accuracy represents the total number of pixels correctly classified divided by the total number of pixels classified as a particular class. IoU is the area of the intersection of the true labels divided by the area of union of the true label and predicted label. A comparison of the validation accuracies during the training process is provided in Fig. 9. The validation accuracy was computed after every 50 epochs. The curves shown in Fig. 9 have been filtered with a moving average filter with a window size of 5 to allow the trends to be easily discernable.

A comparison of pixel accuracy and IoU for the test dataset are provided in Table 5. The results demonstrate that, on average, the multi-task network performs much better than the individual networks. The multi-task network with additive-loss function performs better than the single-task network for all classes with the exception of cracks. The multi-task network with homoscedastic loss performs nominally better than the additive-loss network for identification of material, although the accuracies are very close. These accuracies are documented in Table 5.

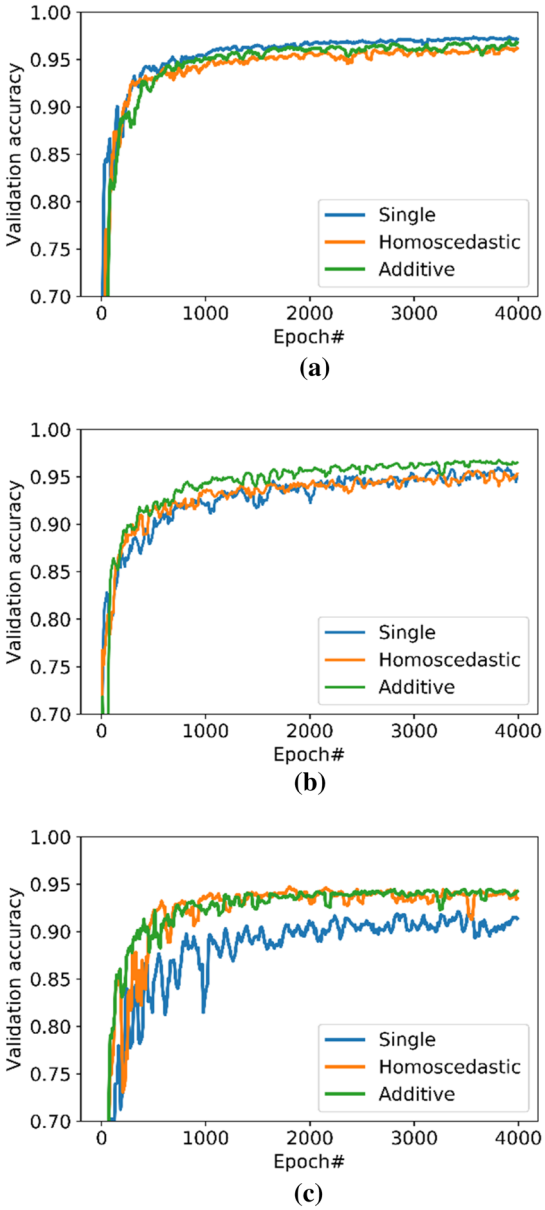

Fig. 9 Validation accuracy comparison for single- and multi-task networks for identification of **a** fine damage, **b** coarse damage, **c** material

4.3 Results from the multi-task network

Sample results from the trained network with the additive-loss function are presented in Fig. 10. The original images were divided into smaller images for training. In the results displayed, the results were upsampled and combined to be the same size as the original image. The upsampling of the network predictions was done using the nearest-neighbor method. The results demonstrate the efficacy of the proposed method for identification of multiple material and damage types. The network is able to identify multiple damage types at different scales and in a variety of combinations (e.g., spalling, crack and exposed reinforcement; spalling and corrosion fatigue cracks and corrosion), demonstrating the versatility of deep learning techniques for local damage and material identification.

Table 5 Test accuracy comparison between single- and multi-task networks

	Pixel accuracy			Intersection over Union (IoU)		
	Single-task	Multi-task		Single-task	Multi-task	
		Homoscedastic	Additive		Homoscedastic	Additive
Material						
Concrete	0.964	**0.985**	0.967	0.879	**0.947**	0.945
Steel	0.871	0.932	**0.940**	0.842	**0.915**	0.911
Asphalt	0.989	0.990	**0.998**	0.974	**0.989**	0.987
Other material	0.462	**0.553**	0.546	0.204	**0.244**	0.236
Mean	0.821	**0.865**	0.863	0.725	**0.774**	0.770
Fine damage						
No damage	**0.977**	0.973	0.972	**0.975**	0.968	0.969
Cracks	**0.964**	0.905	0.949	**0.679**	0.603	0.619
Rebar	**0.873**	0.854	0.861	0.693	0.678	**0.695**
Mean	**0.938**	0.911	0.928	**0.782**	0.749	0.761
Coarse damage						
No damage	0.959	0.960	**0.966**	0.948	0.949	**0.959**
Spalling	0.891	0.936	**0.959**	0.775	0.802	**0.841**
Corrosion	0.952	0.938	**0.958**	0.788	0.753	**0.789**
Mean	0.934	0.944	**0.961**	0.837	0.834	**0.863**
Overall Mean	0.898	0.907	**0.917**	0.781	0.786	**0.798**

Bold values indicate highest accuracy for given class and metric

5 Discussion

The experiments described in the previous section provided several noteworthy observations. This section provides a discussion on five topics, namely (1) accuracy, (2) computational time, (3) feature analysis, (4) homoscedastc vs additive loss function, and (5) use-case scenario.

5.1 Discussion on accuracy

A major performance difference between the multi-task networks and the single-task networks can be seen in the average accuracy. The multi-task networks significantly outperform the single-task networks for material identification and have comparable performance for the identification of damage as evidenced in Table 5. Further insight into the reasons for this improvement can be drawn by looking at predictions of some of the images in the test set.

Sample images where the accuracy between MaDnet and the single network differed significantly are presented in Table 5. Results from all the networks for two different images are shown in Fig. 10. Sample results of automated structural inspection using the multi-task network with additive loss function are shown in Fig. 11. In both these cases, MaDnet is able to identify that the material in the entire image is steel, whereas the single network is unable to do so. One hypothesis is that in the multi-task networks, both the material and damage tasks work together to reinforce the correct prediction. While the damage identification of both networks is similar, the identification of corrosion makes it clear to MaDnet that the material of the beam is steel as opposed to concrete.

5.2 Discussion on computation time

The proposed multi-task architecture affords a significant reduction in computational cost during inference compared to running inference on three single-task networks. The major computational burden for the inference lies in conducting the forward pass through the encoder. As the multi-task network employs a single encoder for all tasks, the computational time is comparable to a single-task network. For all three tasks, MaDnet takes about 0.055 s, whereas the single networks take 0.132 s, resulting in a 58.3% time savings, as shown in Table 6.

5.3 Feature analysis

The ability of the network to generalize depends on the quality of the features learned. These features are guided by the training data and annotations. In the material identification task, due to the wide variety of appearances that concrete and steel members take, training a network for only material identification using a limited size dataset

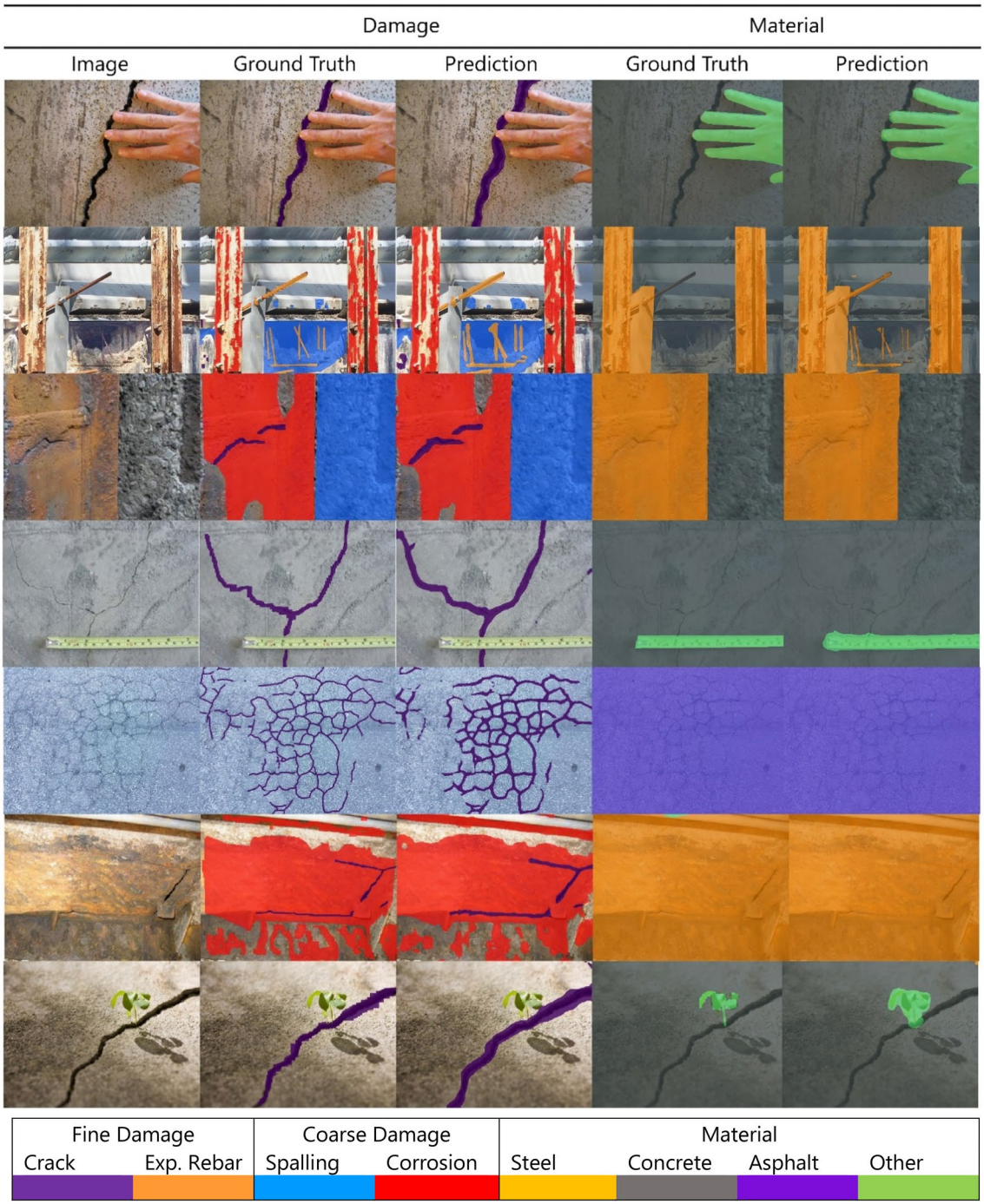

Fig. 10 Sample results of automated structural inspection using the multi-task network with additive-loss function

poses a challenge toward the learning of appropriate features. Fine and coarse damage tasks have more visually consistent patterns and, thus, are more amenable to training a single network. Some handpicked feature maps of the trained networks after the Conv12 layer for the image in Fig. 11b are provided in Fig. 12 to illustrate this difference. While the fine damage, coarse damage, and MaDnet networks have well-trained feature maps, the feature maps of the material network still betray discernable characteristics of the input image. This phenomenon is observed across most of the feature maps after most of the layers. Thus, a reasonable conclusion is that the presence of multiple tasks helps the network learn features suited for all

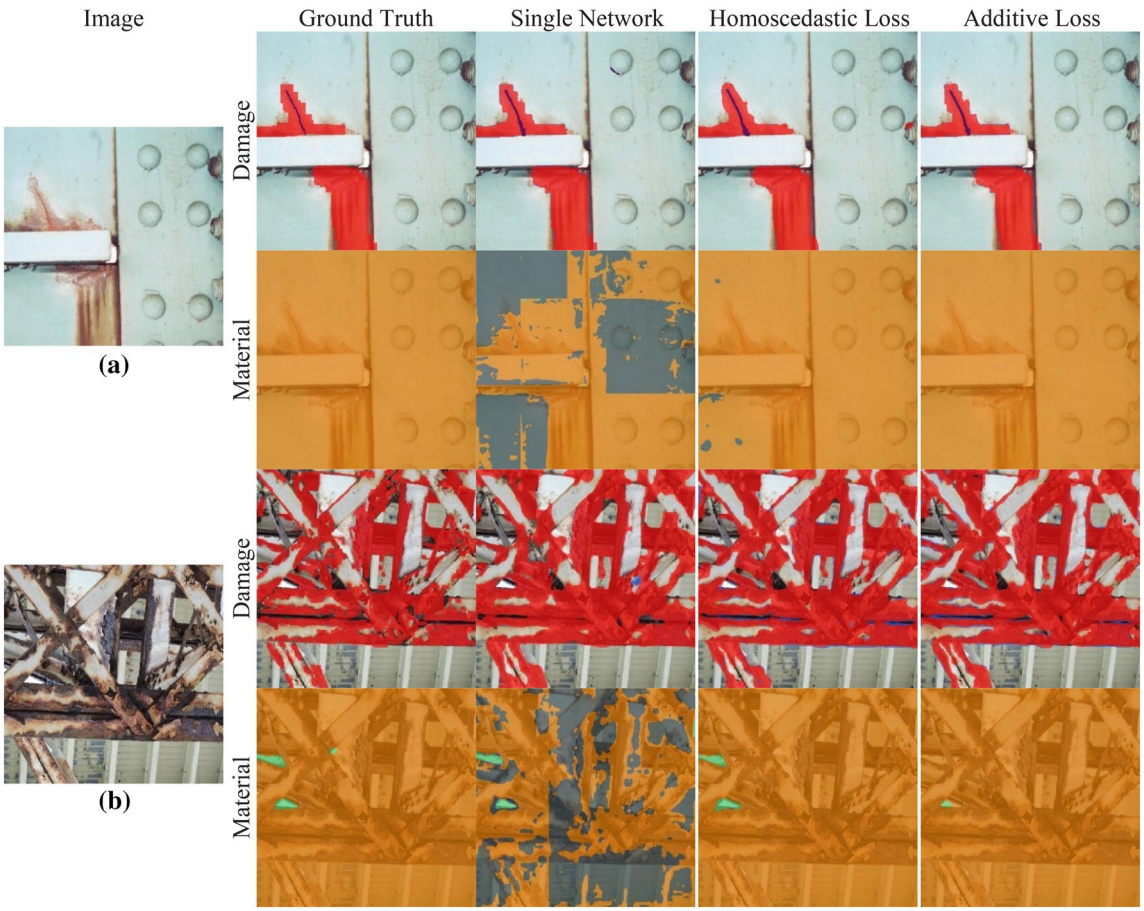

Fig. 11 Comparison of results from single network and multi-task networks

Table 6 Computation time for the networks

Network	Inference time for three tasks (s)
Single networks	0.132
MaDnet	0.055

tasks, thereby increasing the likelihood of high-quality features being learned. In other words, multi-task networks are more likely to learn robust features.

5.4 Homoscedastic vs additive loss

Another interesting phenomenon observed is that MaDnet with additive loss performs nominally better than MaDnet with the homoscedastic loss function for the fine and coarse damage tasks. This is surprising, given that the network with homoscedastic loss function tunes the importance of different loss terms in the combined loss. Figure 13 shows the normalized weight $w_{\text{normal}} = w_{xx} / \sum w_{xx}$ (where $w_{xx} = 1/\sigma_{xx}^2$) for each of the tasks vs. the number of epochs. The normalized weight of the material task, fine damage, and coarse damage tasks gradually converge to about 0.77, 0.16, and 0.7, respectively. The network is thus able to identify the challenging nature of the material damage identification task and tries to compensate for poor performance through increased weighting. This strategy does not necessarily result in higher quality features; instead, the learned features are simply guided by increased weighting for poorly performing tasks. The coarseness of the features of MaDnet with the homoscedastic loss vs. MaDnet with additive loss in Fig. 12 further verifies the hypothesis. The increased robustness of features learned by the additive-loss function is further evidenced by the higher average accuracy in Table 5.

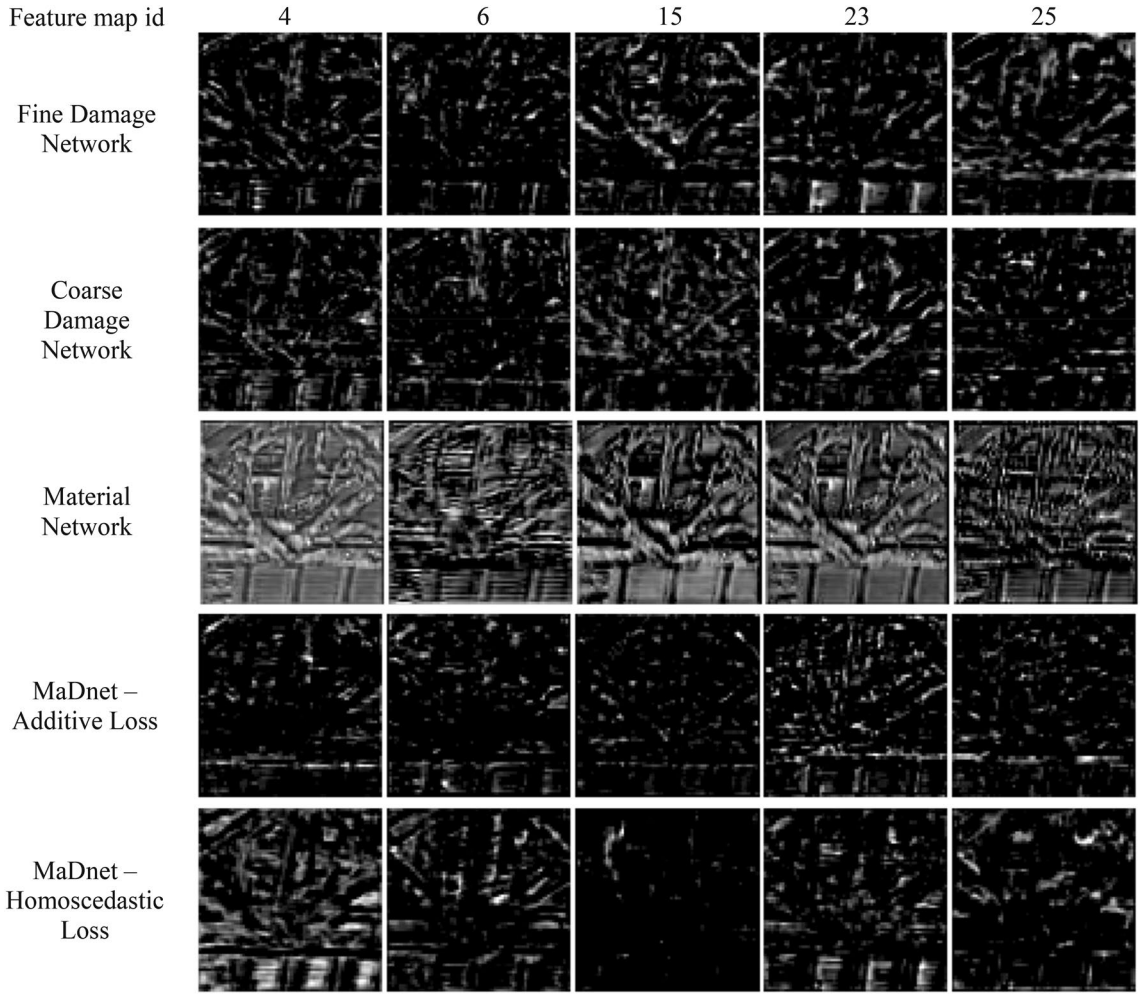

Fig. 12 Feature maps from the Conv12 layer in the encoder

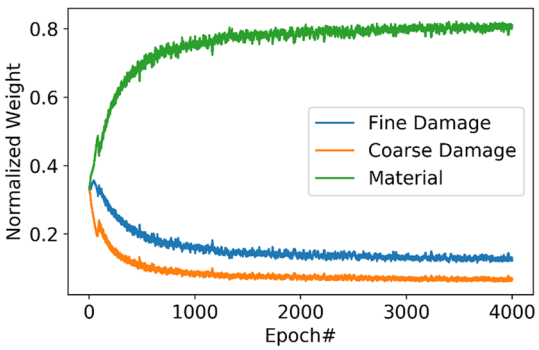

Fig. 13 Normalized homoscedastic weights

5.5 Discussion on use-case scenario

The ability of computer vision techniques to process large amounts of data makes their use in augmenting manual inspections attractive. The fatigue of a manual inspector would increase with the size and number of structures to be inspected, whereas a neural network would be able to process images tirelessly and accurately. Once regions of damage have been identified, manual inspectors may easily look more carefully at the identified regions to make their assessments. Identification of material type along with damage offers the ability to further enhance the inspection process by presenting another parameter for selection of images of interest by an inspector. Knowledge about the material can help to delineate damage that is of structural significance. For example, during a bridge inspection, while spalling, cracks, and corrosion are all important defects to be identified, cracks on steel girders under a deck have different implications than those on

the concrete or asphalt portions of the deck; spalling on concrete and asphalt has different implications. The use of computer vision-based methods will help to streamline the inspection process by increasing the overall efficiency.

6 Conclusions

The paper investigated the semantic segmentation of images for three tasks, namely, the identification of material types, fine damage, and coarse damage. To this end, a multi-task network architecture named MaDnet was proposed to simultaneously conduct all of these tasks and incorporate intrinsic interdependencies. MaDnet is more efficient than individual networks as a single set of features is used to conduct multiple tasks. A new dataset was developed to assess the performance of the proposed network and for the benefit of the wider research community. The proposed networks were tested on the developed dataset and empirically, MaDnet was found to perform better than the single-task networks with an average accuracy of 91.7%, as opposed to 89.8%. MaDnet significantly outperforms the individual networks in the category of material identification and has comparable performance on the other tasks. An important decision to be made while conducting a multi-task optimization is the weighting of the objectives of the individual tasks. Two different combined loss functions were tested, namely a simple additive loss and a homoscedastic loss (an uncertainty weighted objective function where these weights were also learned). Both the additive and homoscedastic loss functions perform well with the additive-loss function having a nominal average performance improvement over the homoscedastic loss. MaDnet also affords a 58.3% reduction in computational time as it requires only one encoder to be processed. An analysis of the feature maps from the inner layers of the networks reveals that multi-task networks like MaDnet are more likely to learn robust features, especially for challenging and data hungry tasks. The incorporation of multiple tasks effectively increases the amount of data available for the network to learn from and improves the robustness of the learned features. The proposed network can be deployed on unmanned aerial vehicles for increased autonomy of structural inspections for numerous inspection applications.

References

1. Koch C, Georgieva K, Kasireddy V et al (2015) A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Adv Eng Inf 29:196–210
2. National Transportation Safety Board (NTSB) (2007) Collapse of I-35W highway bridge Minneapolis, Minnesota August 1, 2007. Highw Accid Rep 178:86
3. ATC (Applied Technology Council) (2005) Field manual: postearthquake safety evaluation of building. ATC-20-1, Redwood City, CA
4. Marroquin A (2017) Inspections bring high degree of difficulty at Hoover Dam bypass bridge. In: Las Vegas Rev. https://www.reviewjournal.com/local/local-nevada/inspections-bring-high-degree-of-difficulty-at-hoover-dam-bypass-bridge/. Accessed 23 Aug 2017
5. Abdel-Qader I, Abudayyeh O, Kelly ME (2003) Analysis of edge-detection techniques for crack identification in bridges. J Comput Civ Eng 17:255–263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)
6. Jahanshahi MR, Kelly JS, Masri SF, Sukhatme GS (2009) A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures. Struct Infrastruct Eng 5:455–486. https://doi.org/10.1080/15732470801945930
7. Yamaguchi T, Hashimoto S (2010) Fast crack detection method for large-size concrete surface images using percolation-based image processing. Mach Vis Appl 21:797–809. https://doi.org/10.1007/s00138-009-0189-8
8. Nishikawa T, Yoshida J, Sugiyama T, Fujino Y (2012) Concrete crack detection by multiple sequential image filtering. Comput Civ Infrastruct Eng 27:29–47. https://doi.org/10.1111/j.1467-8667.2011.00716.x
9. Zhu Z, German S, Brilakis I (2011) Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation. Autom Constr 20:874–883. https://doi.org/10.1016/j.autcon.2011.03.004
10. Paal SG, Jeon J-S, Brilakis I, DesRoches R (2015) Automated damage index estimation of reinforced concrete columns for post-earthquake evaluations. J Struct Eng 141:04014228. https://doi.org/10.1061/(ASCE)ST.1943-541X.0001200
11. Adhikari RS, Moselhi O, Bagchi A (2013) A study of image-based element condition index for bridge inspection. In: ISARC 2013—30th Int Symp Autom Robot Constr Mining, Held Conjunction with 23rd World Min Congr, pp 345–356
12. Chen PH, Shen HK, Lei CY, Chang LM (2012) Support-vector-machine-based method for automated steel bridge rust assessment. Autom Constr 23:9–19. https://doi.org/10.1016/j.autcon.2011.12.001
13. Bonnin-Pascual F, Ortiz A (2014) Corrosion detection for automated visual inspection. Dev Corros Prot. https://doi.org/10.5772/57209
14. Yeum CM, Dyke SJ (2015) Vision-based automated crack detection for bridge inspection. Comput Civ Infrastruct Eng 30:759–770. https://doi.org/10.1111/mice.12141
15. Hu Y, Zhao C (2010) A novel LBP based methods for pavement crack detection. J Pattern Recognit Res 5:140–147. https://doi.org/10.13176/11.167
16. Zhang W, Zhang Z, Qi D, Liu Y (2014) Automatic crack detection and classification method for subway tunnel safety monitoring. Sensors (Switzerl) 14:19307–19328. https://doi.org/10.3390/s141019307
17. Wu S, Zhong S, Liu Y (2017) Deep residual learning for image steganalysis. Multimed Tools Appl 2017:1–17
18. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. Int Conf Learn Represent. https://doi.org/10.1016/j.infsof.2008.09.005
19. LeCun Y, Bengio Y, Hinton G et al (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
20. Banga S, Gehani H, Bhilare S, et al (2018) 3D Topology Optimization using Convolutional Neural Networks. arXiv preprint arXiv:1808.07440v1

21. Lei X, Liu C, Du Z et al (2019) Machine learning-driven real-time topology optimization under moving morphable component-based framework. J Appl Mech Trans ASME. https://doi.org/10.1115/1.4041319
22. Sosnovik I, Oseledets I (2019) Neural networks for topology optimization. Russ J Numer Anal Math Model 34:215–223. https://doi.org/10.1515/rnam-2019-0018
23. Hoskere V, Eick BA, Spencer BF, Smith MD, Foltz SD (2019) Deep Bayesian neural networks for damage quantification in miter gates of navigation locks. Struct Heal Monit. https://doi.org/10.1177/1475921719882086
24. Rafiei MH, Adeli H (2018) A novel unsupervised deep learning model for global and local health condition assessment of structures. Eng Struct 156:598–607. https://doi.org/10.1016/j.engstruct.2017.10.070
25. Bao Y, Tang Z, Li H, Zhang Y (2019) Computer vision and deep learning–based data anomaly detection method for structural health monitoring. Struct Heal Monit 18:401–421. https://doi.org/10.1177/1475921718757405
26. Ye XW, Jin T, Yun CB (2019) A review on deep learning—based structural health monitoring of civil infrastructures. Smart Struct Syst 24:567–586. https://doi.org/10.12989/sss.2019.24.5.567
27. Tsuchimoto K, Narazaki Y, Hoskere V, Spencer Jr. BF (2020) Rapid postearthquake safety evaluation of buildings using sparse acceleration measurements. Struct Heal Monit (in press)
28. Dorafshan S, Maguire M (2018) Bridge inspection: human performance, unmanned aerial systems and automation. J Civ Struct Heal Monit 8:443–476. https://doi.org/10.1007/s13349-018-0285-4
29. Chen F-C, Jahanshahi MR, Johnson D, Delp EJ (2019) Vision-based decision support for flood risk assessment using google street view images. Struct Heal Monit. https://doi.org/10.12783/SHM2019/32472
30. Alipour M, Harris DK (2020) A big data analytics strategy for scalable urban infrastructure condition assessment using semi-supervised multi-transform self-training. J Civ Struct Heal Monit 10:313–332. https://doi.org/10.1007/s13349-020-00386-4
31. Yeum CM, Dyke SJ, Ramirez J (2018) Visual data classification in post-event building reconnaissance. Eng Struct 155:16–24. https://doi.org/10.1016/J.ENGSTRUCT.2017.10.057
32. Dimitrov A, Golparvar-Fard M (2014) Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections. Adv Eng Informatics 28:37–49. https://doi.org/10.1016/j.aei.2013.11.002
33. Roberts D, Torres Calderon W, Tang S, Golparvar-Fard M (2020) Vision-based construction worker activity analysis informed by body posture. J Comput Civ Eng 34:04020017. https://doi.org/10.1061/(asce)cp.1943-5487.0000898
34. Fang Q, Li H, Luo X et al (2018) A deep learning-based method for detecting non-certified work on construction sites. Adv Eng Informatics 35:56–68. https://doi.org/10.1016/j.aei.2018.01.001
35. Kim H, Kim H, Hong YW, Byun H (2018) Detecting construction equipment using a region-based fully convolutional network and transfer learning. J Comput Civ Eng 32:04017082. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000731
36. Yeum CM, Choi J, Dyke SJ (2017) Autonomous image localization for visual inspection of civil infrastructure. Smart Mater Struct 26:035051. https://doi.org/10.1088/1361-665X/aa510e
37. Narazaki Y, Hoskere V, Hoang TA, Spencer Jr. BF (2017) Vision-based automated bridge component recognition integrated with high-level scene understanding. In: The 13th international workshop on advanced smart materials and smart structures technology (ANCRiSST), Tokyo, Japan
38. Narazaki Y, Hoskere V, Hoang TA et al (2019) Vision-based automated bridge component recognition with high-level scene consistency. Comput Civ Infrastruct Eng mice 2019:12505. https://doi.org/10.1111/mice.12505
39. Narazaki Y, Hoskere V, Hoang TA, Spencer Jr. BF (2018) Automated bridge component recognition using video data. In: The 7th world conference on structural control and monitoring, 7WCSCM
40. Zhang L, Yang F, Daniel Zhang Y, Zhu YJ (2016) Road crack detection using deep convolutional neural network. IEEE Int Conf Image Process 2016:3708–3712. https://doi.org/10.1109/ICIP.2016.7533052
41. Yeum CM (2016) Computer vision-based structural assessment exploiting large volumes of images. Theses Diss Available from ProQuest
42. Hoskere V, Narazaki Y, Hoang TA, Spencer BF (2017) Vision-based Structural Inspection using Multiscale Deep Convolutional Neural Networks. In: 3rd Huixian International Forum on Earthquake Engineering for Young Researchers, University of Illinois, Urbana-Champaign. Urbana, Il, USA
43. Cha YJ, Choi W, Suh G et al (2018) Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput Civ Infrastruct Eng 33:731–747. https://doi.org/10.1111/mice.12334
44. Rubio JJ, Kashiwa T, Laiteerapong T et al (2019) Multi-class structural damage segmentation using fully convolutional networks. Comput Ind 112:103121. https://doi.org/10.1016/j.compind.2019.08.002
45. Spencer BF, Hoskere V, Narazaki Y (2019) Advances in computer vision-based civil infrastructure inspection and monitoring. Engineering 5:199–222
46. FHWA (2004) National bridge inspection standards regulations (NBIS). Fed Regist 69:15–35
47. Caruana R (1997) Multitask learning. Mach Learn 28:41–75. https://doi.org/10.1023/A:1007379606734
48. Kendall A, Gal Y, Cipolla R (2017) Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. Mach Learn. https://doi.org/10.1109/CVPR.2018.00781
49. Teichmann M, Weber M, Zöllner M et al (2018) MultiNet: real-time joint semantic reasoning for autonomous driving. In: IEEE intelligent vehicles symposium, proceedings. institute of electrical and electronics engineers Inc., pp 1013–1020
50. Moeskops P, Wolterink JM, van der Velden BHM et al (2016) Deep learning for multi-task medical image segmentation in multiple modalities. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). Springer, pp 478–486
51. Facial Landmark Detection by Deep Multi-task Learning (2019) https://mmlab.ie.cuhk.edu.hk/projects/TCDCN.html. Accessed 15 Nov 2019
52. Shelhamer E, Long J, Darrell T et al (2015) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 2015:640–651
53. Krogh A, Hertz JA (1992) A simple weight decay can improve generalization. Adv Neural Inf Process Syst 4:950–957
54. Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. Int Conf Learn Represent 2015:1–15. https://doi.org/10.1145/1830483.1830503
55. Abadi M, Barham P, Chen J et al (2016) TensorFlow: a system for large-scale machine learning. In: 12th symposium on operating systems design and implementation (2016). GA, pp 265–283
56. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 580–587
57. Hoskere V, Narazaki Y, Hoang TA, Spencer Jr. BF (2017) Vision-based structural inspection using multiscale deep convolutional neural networks. In: 3rd Huixian International Forum on

Earthquake Engineering for Young Researchers, University of Illinois, Urbana-Champaign
58. Chungwook S, Enrique V, Jhon PS et al (2016) Performance of low-rise reinforced concrete buildings in the 2016 Ecuador earthquake. https://datacenterhub.org/resources/14160. Accessed 6 Aug 2017
59. Bridgehunter.com: Historic Bridges of the United States (2017) https://bridgehunter.com/. Accessed 6 Aug 2017
60. US Army Corps of Engineers (2017) https://www.usace.army.mil/. Accessed 6 Aug 2017
61. Farabet C, Couprie C, Najman L, Lecun Y (2013) Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell 35:1915–1929. https://doi.org/10.1109/TPAMI.2012.231
62. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE transactions on pattern analysis and machine intelligence, pp 3431–3440
63. Zhang S, Fu H, Yan Y et al (2019) Attention guided network for retinal image segmentation. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 11764:797–805. https://doi.org/10.1007/978-3-030-32239-7_88