Noname manuscript No.
(will be inserted by the editor)

Localizing and Quantifying Infrastructure Damage Using Class Activation Mapping Approaches

Xukun Li · Doina Caragea · Huaiyu Zhang · Muhammad Imran

Received: date / Accepted: date

Abstract Traditional post-disaster assessment of damage heavily relies on expensive geographic information system (GIS) data, especially remote sensing image data. In recent years, social media has become a rich source of disaster information that may be useful in assessing damage at a lower cost. Such information includes text (e.g., tweets) or images posted by eyewitnesses of a disaster. Most of the existing research explores the use of text in identifying situational awareness information useful for disaster response teams. The use of social media images to assess disaster damage is limited. We have recently proposed a novel approach, based on convolutional neural networks and class activation mapping, to locate building damage in a disaster image and to quantify the degree of the damage. In this paper, we study the usefulness of the proposed approach for other categories of infrastructure damage,

X. Li
Department of Computer Science
Kansas State University
Manhattan, KS 66502
E-mail: xukun@ksu.edu

D. Caragea
Department of Computer Science
Kansas State University
Manhattan, KS 66502
E-mail: dcaragea@ksu.edu

H. Zhang
Department of Statistics
Kansas State University
Manhattan, KS 66502
E-mail: huaiyu@ksu.edu

M. Imran
Qatar Computing Research Institute
Hamad Bin Khalifa Universit
Doha, Qatar
E-mail: mimran@hbku.edu.qa

specifically bridge and road damage, and compare two class activation mapping approaches in this context. Experimental results show that our proposed approach enables the use of social network images for post-disaster infrastructure damage assessment, and provides an inexpensive and feasible alternative to the more expensive GIS approach.

Keywords Image analysis · convolutional neural networks (CNN) · class activation mapping (CAM) approaches · damage localization · bridge, building and road damage

1 Introduction

Fast detection of damaged areas after an emergency event can inform responders and aid agencies, support logistics involved in relief operations, accelerate real-time response, and guide the allocation of resources. Most of the existing studies on detecting and assessing disaster damage rely heavily on macro-level images, such as remote sensing imageries [37,14], [11], or imageries transmitted by unmanned aerial vehicles [3]. Collection and analysis of macro-level images require costly resources, including expensive equipment, complex data processing tools, and also good weather conditions. To benefit the response teams, the macro-level images have to be collected and analyzed very fast, which is not always possible with traditional collection and analysis methods.

With the growth of social media platforms in recent years, real-time disaster-related information is readily available, in the form of network activity (e.g., number of active users, number of messages posted), text (e.g., tweets), and images posted by eyewitnesses of disasters on platforms such as Twitter, Facebook, Instagram or Flicker. Many studies have shown the utility of social media information for disaster management and response teams. For example, the analysis of text data (e.g., tweets from Twitter) has received significant attention in recent works [13], [17], [19], [22], [39]. However, social media images, while very informative [4], have not been extensively used to aid disaster response, primarily due to the complexity of information extraction from (noisy) images, as compared to information extraction from text.

By contrast with macro-level images, social media images have higher "resolution", in the sense that they can provide detailed on-site information from the perspective of the eyewitnesses of the disaster [4]. Thus, social media images can serve as an ancillary yet rich source of visual information in disaster damage assessment. Pioneering works with focus on the utility of social media images in disaster response include [1], [28], where the goal is to use convolutional neural networks (CNN) to assess the severity of the damage (specifically, to classify social media images based on the degree of the damage as: *severe*, *mild*, and *none*).

Due to ground-breaking developments in computer vision, many image analysis tasks have become possible. Disaster management and response teams can benefit from novel image analyses that can produce quantitative assessments of damage, and inform the relief operations with respect to priority areas. In this context, it is useful to locate damage areas in social media images (when images contain damage), and subsequently use the identified damage areas to assess the damage severity on a continuous scale. Possible approaches for localizing damage in social media images include object detection [7,25] and image segmentation [36]. Object detection can be conducted by classifying some specific regions in an image as containing damage or not. For example, Cha et al. [7] used a convolution neural network (CNN) to classify small image regions (with 256×256 pixel resolutions) as containing concrete crack damage or not. Maeda et al. [25] used a state-of-the-art object detection approach, called Single Shot MultiBox Detector (SSD) [24], to detect several types of road damage. Image segmentation has been used in [36] to detect building damage based on high resolution aerial images.

Regardless of the method used, object detection or image segmentation, existing approaches for localizing damage first identify objects (i.e., potential damage regions) and subsequently classify the objects as *damage* (sometimes, *severe* or *mild*) or *no damage*. Thus, there is a conceptual mismatch in the way existing approaches are used, given that damage is generally regarded as a high-level concept rather than a well-defined object. By first identifying objects and then assigning discrete hard-labels to them, existing approaches produce a clear-cut boundary for the damaged areas, although a smooth boundary would be more appropriate.

To address this limitation, we have recently proposed a novel approach [23], called Damage Detection Map (DDM), to generate a smooth damage heatmap for an image. Our approach adopts the gradient-weighted CAM (Grad-CAM) [32] technique to localize the area in an image which contributes to the damage class. Based on the damage heatmap, we also propose a new quantitative measure, called Damage Assessment Value (DAV), to quantify the severity of damage on a continuous scale. Our approach was originally tested on a set of images that show building damage. In this paper, we evaluate the usefulness of the proposed approach for localizing and quantifying different types of infrastructure damage, including building damage, bridge damage and road damage. Furthermore, in addition to Grad-CAM, we also use its Grad-CAM++ extension [8], which can identify multiple occurrences of a class object, and compare Grad-CAM and Grad-CAM++ in the context of damage localization and quantification.

Extensive experimental results show that our approach makes the disaster infrastructure damage localization possible, and thus extends the use of social media images in disaster assessment.

The rest of this paper is organized as follows: We discuss related work in Section 2, and describe the proposed approaches for generating DDM heatmaps, and computing DAV scores in Section 3. We describe the experimental setup in Section 4 and present the results in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

Social media data has been shown to have significant value in disaster response [6,26,30]. Many machine learning approaches [2,17,22], including deep learning approaches [5,27], have been proposed and used to help identify and prioritize useful *textual* information (e.g., tweets) in social media. Some works have focused specifically on identifying situational awareness information [33,16], including information related to damage assessment [19,39,31,9].

Despite the extensive use of machine learning tools for analyzing social media text data posted during disaster events, there is not much work on analyzing social media images posted by eyewitnesses of a disaster. One pioneering work in this area [28], used trained CNN models, specifically, VGG16 networks fine-tuned on disaster image datasets, and showed that the CNN models perform better than standard techniques based on bags-of-visual-words.

Other prior works focused on image-based disaster damage assessment use aerial or satellite images, e.g. [37,14]. Compared to such works, which use more expensive imagery, we focus on the use of social media images, which are readily available during disasters, together with interpretability approaches, i.e., Grad-CAM, to produce a damage map and a damage severity score for each image.

Similar to us, Nia and Mori [29] use ground-level images collected using Google to assess building damage. Their model consists of three different CNN networks (fine-tuned with raw images, color-masked or binary-masked images, respectively) to extract features predictive of damage. Subsequently, a regression model is used with the extracted features to predict the severity of the damage on a continuous scale. Compared to [29], we use the features identified at the last convolutional layer of the CNN network to build a detection map, and use the map to produce a numeric damage severity score. While CAM-type approaches have been used to explain model predictions in many other application domains, to the best of our knowledge, they have not been used to locate damage and assess damage severity in prior work.

3 Methods

Our approach generates a Damage Detection Map, which visualizes the damage area for a given image, and a score DAV, which quantifies the severity of the damage. The main components of our approach, shown in Fig. 1, are the following: 1) a CNN that classifies images into several classes, e.g., *building-damage* and *no-building-damage*, or *building-damage*, *bridge-damage*, *road-damage* and *no-damage*; 2) a class activation mapping (specifically, Gradient-weighted Class Activation Mapping, or Grad-CAM, and Grad-CAM++), which generates the DDM map by weighting the last convolutional layer of the CNN model; 3) finally, the damage severity score computed by averaging the values in the map. The details for the three components of our approach are provided in what follows.

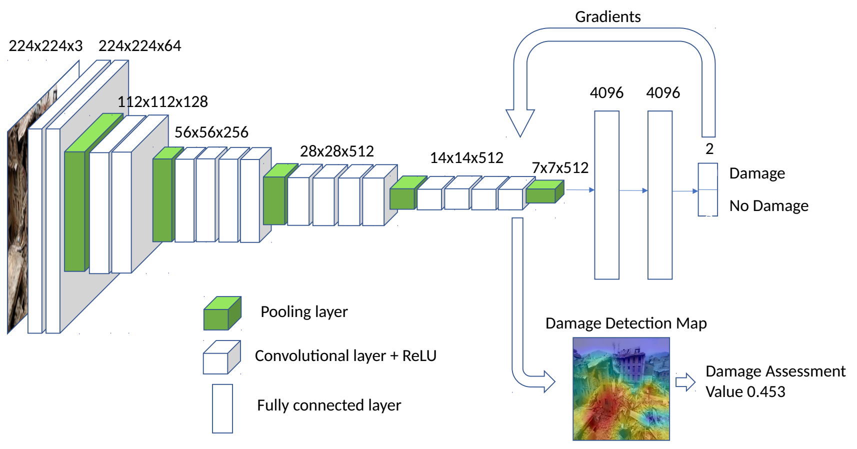


Fig. 1: Overview of the proposed approach, when learning to discriminate between *damage* and *no-damage* images. The model takes an image as input and produces a Damage Detection Map (i.e., heatmap) and a Damage Assessment Value as output. The model shown is a binary model that learns to discriminate between images showing damage and images not showing damage. However, the model can be generalized to multiple classes (e.g., types of damage) by changing the number of units in the output (last) layer.

3.1 Convolutional Neural Networks

Convolutional neural networks [21] have been used successfully for many image analysis tasks [20,12]. A Convolutional Neural Network (CNN) takes an image as input and outputs the class that the image belongs to. It consists of convolutional layers, together with non-linear Rectified Linear Unit (ReLU) activations, pooling layers and fully-connected layer. Convolution layers use learnable filters to identify predictive features in input images. A filter is convolved (slided) across an input image, and a new image is produced by computing dot products between the filter and the current window of the input image. ReLU activation [18] is widely used with modern neural networks, as they it does not activate all neurons at the same time. Different filters can potentially identify different features in the input image. Pooling layers are used to reduce the number of parameters, and help increase the generalizing ability of the network and avoid overfitting. In particular, max-pooling reduces the number of parameters by iteratively taking the maximum value in a local area of the matrix corresponding to its input images. A fully connected layer connects all its units to units in the next layer.

The ImageNet annual competition (where a dataset with 1.2 million images in 1000 categories is provided to participants) has led to several popular architectures, including AlexNet [18], VGG19 [34], ResNet [15] and Inception [35]. We choose VGG19 as the architecture for our CNN model, as VGG19 has good classification accuracy and it is relatively simpler compared to ResNet and Inception. Furthermore, pre-trained models are available for VGG19 and it is easy to fine tune them for different classification problems.

VGG19 [34] contains 16 convolutional layers (with 5 pooling layers) and 3 fully connected layers. Each convolution layer is equipped with a non-linear ReLU activation [18]. The convolutional layers can be seen as feature extraction layers, where each successive layer detects predictive image features (i.e., image fragments that correspond to edges, corners, textures, etc.) at a more abstract level than the previous layer.

As can be seen in Fig. 1, the size of the input to the convolutional layers is reduced by a factor of 2 through max-pooling layers, but not all convolution layers are followed by a max-pooling layer. After every max-pooling layer, the width of the convolution layer (i.e., number of filters used) increases by a factor of 2. After the last max-pooling layer, there are two fully connected layers with dimension 4096, and another fully connected layer whose neurons correspond to the categories to be assigned to the input image. The last layer of the standard VGG19 model has dimension 1000 because VGG19 was originally trained on a dataset with 1000 categories. However, as we are interested in using VGG19 to classify images in two or more categories, e.g., *building-damage* and *no-building-damage*, we change the dimension of the last fully connected layer from 1000 to the number of categories used for training (in the *building-damage* versus *no-building-damage* model, the number of categories is 2). Overall, the model includes more than 130 million parameters, and it takes a significant amount of time (and a large number of images) to train it accurately [34]. However, the model parameters are highly transferable to other image classification problems [38]. Thus, to avoid the need for a large number of images, we initialize our model with the pre-trained VGG19 model (except for the last fully connected layer), and fine tune it using disaster-related images. More specifically, given a training image x with label y represented as a one-hot vector (e.g., if the label is *damage*, then $y = [1, 0]$, otherwise $y = [0, 1]$), all the parameters θ will be updated by:

$$\theta \longleftarrow \theta - \mu \frac{\partial \mathcal{L}(y, \text{CNN}(x))}{\partial \theta}, \tag{1}$$

where μ is learning rate, $\mathcal{L}$ is the cross-entropy loss, and $\text{CNN}(x)$ is the output of the CNN given input x.

3.2 Class Activation Mapping Approaches

Recent works have focused on approaches that can explain the results of the deep learning models. Such approaches include Gradient-weighted Class Activation Mapping (Grad-CAM) [32] and its extension Grad-CAM++ [8]. In a general classification problem, for an input image and a trained CNN model, Grad-CAM makes use of the gradients of a target category to compute a category-specific weight for each feature map of a convolution layer. The weights are used to aggregate the feature maps of the final convolutional layer, under the assumption that the last level captures the best trade-off between high-level semantic features and spatial information. The resulting maps can be used to identify the discriminative regions for the target category (which explain the CNN model's prediction), and implicitly to localize the category in the input image. Thus, Grad-CAM can be seen as a weakly supervised approach, which can localize a category in an image based only on global image labels [32]. Furthermore, the Grad-CAM localized categories or objects (shown using heatmaps) have soft boundaries, and can be used to gain both insight and trust into the model.

Chattopadhyay et al. [8] identify two main drawbacks of the Grad-Cam approach, and proposed an extension, called Grad-CAM++, to address the Grad-CAM drawbacks. First, they observed that Grad-CAM does not properly identify/localize all occurrences of a class object. Furthermore, Grad-CAM may not always localize the whole class object, but only parts of it.

We formally describe the Grad-CAM [32] and the Grad-CAM++ [8] approaches in the remaining of this subsection.

Let f^k, for $k = 1, \ldots, 512$, be a feature map in the last convolutional layer (of dimension $14 \times 14 \times 512$) of the VGG19 network (see Fig. 1). Each $f^k_{(i,j)}$ represents the value at location (i, j) in the k-th feature map, for $i = 1, \ldots, 14$, $j = 1, \ldots, 14$, $k = 1, \ldots, 512$. Let $\tilde{y}_c$ be the numeric output score corresponding to a category of interest y_c (e.g., damage) just before the softmax function (which transforms the score $\tilde{y}_c$ to a binary 0/1 value) is applied. Given an input image x and the score $\tilde{y}_c$, we define the weights w_k corresponding to the feature maps f^k as the sum of the gradients of the score $\tilde{y}_c$ with respect to $f^k_{(i,j)}$, for all i, j. Specifically:

$$w_k = \frac{1}{14 \times 14} \sum_{i,j} \frac{\partial \tilde{y}_c}{\partial f^k_{(i,j)}} \tag{2}$$

For the feature maps f^k and their corresponding weights w_k, we finally define a 14×14 matrix S, such that

$$s_{i,j} = \text{ReLU}\left(\sum_k w_k f^k_{(i,j)}\right) \tag{3}$$

The ReLU function in Equation (3) is used to cancel the effect of the negative values, while emphasizing the effect of the positive values. We should note that the feature maps f^k in Equations (2) and (3) are in the last convolutional layer of the VGG19 network, as this layer generally shows a good trade-off between high-level features and spatial information in the original image, and can thus lead to good explanations for the classifications made by the model.

However, Chattopadhyay et al. [8] noted that if a target class object has multiple occurrences in an image, marked by footprints with different orientations and sizes, then only the object occurrences with larger footprint will be visible in the heatmaps produced by Grad-CAM. To alleviate this problem, they proposed Grad-CAM++, which assigns weighting coefficients $\alpha_{i,j}^{k}$ to the pixel (i, j) gradients for target category score $\tilde{y}_c$ and feature map k. Thus, in Grad-CAM++, Equation (2) becomes:

$$w'_k = \sum_{i,j} \alpha_{i,j}^{k} ReLU \left(\frac{\partial \tilde{y}_c}{\partial f_{(i,j)}^{k}} \right) \tag{4}$$

The effect of the ReLU function in Equation (4) is similar to the effect of ReLU in Equation (3): it helps emphasize the positive values that contribute to the importance (i.e., weight) of a particular feature map. Chattopadhyay et al. [8] derived closed-form formulas for calculating the pixel-wise coefficients $\alpha_{i,j}^{k}$ for a feature map f_k and a target category y_c. Replacing w_k with w'_k in Equation (3) leads to higher weights for features maps corresponding to target objects with a smaller footprint in the original images, and thus allows for multiple occurrences of the object to be identified (and similarly allows for the localization of an object in its entirety).

We will experiment with both Grad-CAM and Grad-CAM++ approaches to identify and localize damage in disaster images.

3.3 Damage Detection Map

Our proposed Damage Detection Map (DDM) is inspired by Class Activation Mapping approaches. As described in the previous section, Grad-CAM/Grad-CAM++ identify target class objects and localize them using heatmaps. The heatmaps are simply created based on the class labels of the original images, and show the target objects using soft-boundaries. This makes CAM-based approaches particularly attractive for the problem of localizing damage in disaster images, assuming that only coarse labels of images as *damage* (or specific type of damage, e.g., *bridge-damage*, *road-damage*, *building-damage*) and *no-damage* are available for training. We would like to emphasize that the heatmaps showing categories of interest using soft-boundaries are very appropriate for localizing damage, as damage boundaries are inherently soft. Moreover, the heatmaps that explain the model's predictions can be used to gain the trust of disaster management teams, and thus increase the usability of social media images in disaster response and recovery.

Given an input disaster image x and a category of interest (e.g., *building-damage*), we use the Grad-CAM and Grad-CAM++ to produce a 14×14 matrix S (and correspondingly a 14×14 image) as described in the previous section. Using a bilinear interpolation technique, we further resize S to S_C to match the dimensions of the input image. The same resizing method and dimensions were used in Grad-CAM to generate heatmaps for explaining the predictions of a CNN [32], and have been widely adopted in the literature. The heatmap showing the S_C values is the final Damage Detection Map, and can be used to visualize regions of the image that are discriminative with respect to the category of interest.

3.4 Quantifying Damage Severity

When a disaster occurs, eyewitnesses of the disaster will produce a huge number of images in a short period of time. An important goal of disaster assessment is to extract and concisely summarize the information contained in the images posted by eyewitnesses. To meet this demand, we propose to use a damage assessment value (DAV) derived from the DDM heatmap to represent the damage severity for each image.

The disaster damage map uses numerical values to measure the intensity of each pixel of the image (the higher the intensity, the more severe the damage), and can be represented as a heatmap. We take the average over all the numerical values in the heatmap of a given image, and use the resulting value as an overall score for the severity of the damage. Formally, we define the damage assessment value (DAV) as:

$$DAV = \frac{1}{14 \times 14} \sum_{i,j} s_{i,j} \tag{5}$$

where $s_{i,j}$ are the elements of the S matrix defined in Equation (3), and 14×14 is the dimension of the matrix S.

4 Experimental Setup

We perform a series of experiments to evaluate the performance of our proposed method. The experiments are designed to answer the following questions:

1. Can the Damage Detection Map accurately locate the damage areas?
2. What type of damage can be localized more accurately?
3. Can the DAV scores provide a reliable measure for damage severity?
4. Among two-class (*damage* or *no-damage*), four-class (*building-damage*, *bridge-damage*, *road-damage*, or *no-damage*) or six-class (*building-damage*, *no-building-damage*, *bridge-damage*, *no-bridge-damage*, *road-damage*, or *no-road-damage*) models, what model leads to better results overall?
5. Between Grad-CAM and Grad-CAM++, which approach is better?

4.1 Data Description

The main dataset we used in this paper was originally published in [28] and it is available from `http://crisisnlp.qcri.org`. The dataset was assembled from Google by using search queries such as *building-damage*, *bridge-damage*, *road-damage*. In addition to images in one of these three categories, the dataset contains images labeled as *no-damage*. The numbers of images in the *building-damage*, *bridge-damage*, *road-damage* and *no-damage* categories are 903, 813, 826, 463, respectively. From the original dataset, we manually filtered out

Category	Bridge	Building	Road	Total
Damage	347	709	525	1581
No-Damage	731	510	497	1738
Total	1078	1219	1022	3319

Table 1: Statistics about the Google damage dataset, which contains images in the following categories: *bridge-damage*, *no-bridge-damage*, *building-damage*, *no-building-damage*, *road-damage*, and *no-road-damage*

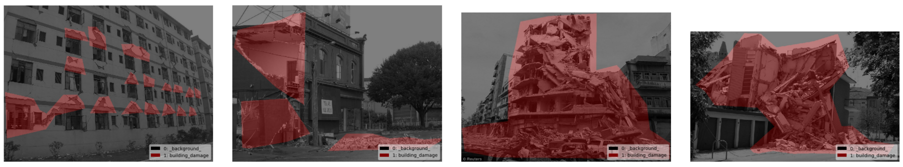
images that did not contain buildings, bridges or roads, and also noisy/mislabeled images. Furthermore, we manually classified the remaining *no-damage* images as *no-building-damage*, *no-bridge-damage* and *no-road-damage*, and used Google search to crawl more images in these specific no-damage categories. Finally, as we aim to localize damage in images, we manually marked the damage area in each damage image. As damage is not an object, heatmaps with smooth boundaries are preferable to bounding boxes when localizing damage. However, human annotators cannot provide precise heatmaps, unless they have professional knowledge of disaster damage, in which case their annotation would be very expensive. To reduce the cost, generally human annotators will simply mark the regions of an image that contain damage (resulting in a binary included/not included representation). We used the tool LabelMe, available from `http://labelme.csail.mit.edu`, to mark the damage.

We should note that the task of localizing damage in an image is very subjective, as different people may see things differently in a damaged area. While subjective, the datasets we created, which will be made available to the research community, can be used to gain insights into computational approaches and challenges posed by the nature of the data. Meanwhile, a better dataset could potentially be assembled by collaborating with infrastructure damage experts in the future. The statistics about the final dataset used are shown in Table 1. Sample images in the damage categories, together with the ground-truth localization annotations, are shown in Fig. 2. We will refer to this dataset as *Google dataset* in what follows.

We randomly split the Google dataset into training (80%) and test (20%) subsets. The VGG19 models were fine-tuned on the training subset, and the evaluation was performed on the test subset.

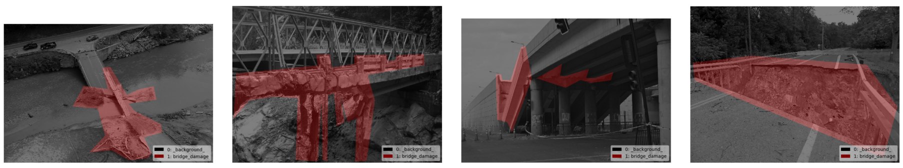
To evaluate the ability of the DAV scores to capture the severity of the damage in an image, we used the dataset published in [29]. This dataset contains images with building damage caused by natural disasters. Each image was manually labeled using one of the following categories, which are indicative of the degree of damage: *no-damage*, *slight-damage*, *moderate-damage*, *heavy-damage*, and *total-destruction*. Each category is associated with a numeric value: *no-damage* $\leftrightarrow$ 0, *slight-damage* $\leftrightarrow$ 0.25, *moderate-damage* $\leftrightarrow$ 0.5, *heavy-damage*$\leftrightarrow$ 0.75, *total-destruction* $\leftrightarrow$ 1. The statistics about the final dataset used are shown in Table 2. Sample images with different degrees of severity, are shown in Fig. 3. We will refer to this dataset as *Building Damage Severity* dataset in what follows.

(a) Original *building-damage* images

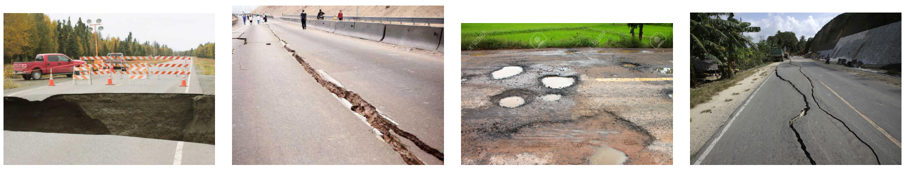


(b) Annotated *building-damage* images

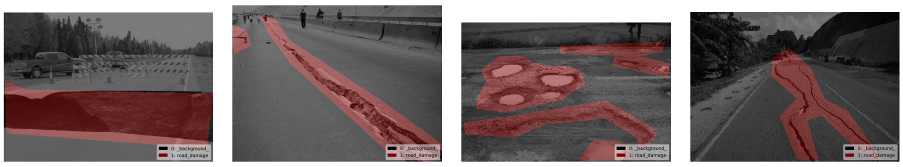
(c) Original *bridge-damage* images



(d) Annotated *bridge-damage* images



(e) Original *road-damage* images



(f) Annotated *road-damage* images

Fig. 2: Sample images together with their corresponding damage localization annotation for *building-damage*, *bridge-damage* and *road-damage*, respectively.

Severity	0	0.25	0.5	0.75	1	Total
Damage	46	15	47	86	56	250

Table 2: Statistics about the Building Damage Severity dataset, which contains building images in the following categories: *no-damage* (0), *slight-damage* (0.25), *moderate-damage* (0.5), *heavy-damage* (0.75), and *total-destruction* (1)

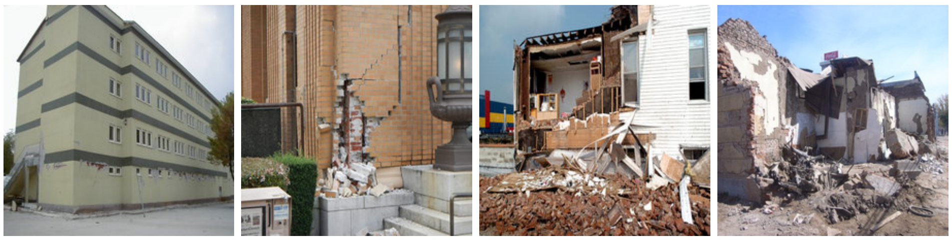


Fig. 3: Sample images from the Building Damage Severity dataset, together with their severity annotations

4.2 VGG19 Models Trained

We trained two-class, four-class and six-class VGG19 models. There are three two-class models trained to discriminate between *damage* and *no-damage* images, specifically, *building-damage* versus *no-building-damage*, *bridge-damage* versus *no-bridge-damage*, and *road-damage* versus *no-roda-damage*, respectively. The four-class model was trained to discriminate between *road-damage*, *building-damage*, *bridge-damage* and *no-damage* images. Finally, the six-class model was trained to discriminate between *building-damage*, *no-building-damage*, *bridge-damage*, *no-bridge-damage*, *road-damage*, and *no-road-damage*.

4.2.1 Hyper-parameters

We used TensorFlow's GradientDescentOptimizer to train the model using mini-batch gradient descent on a GeForce GTX 1070 graphic card. Based on preliminary experimentation, we chose to use a learning rate of 0.001 and a batch size of 32 images in all our experiments. Furthermore, we used the dropout technique with a rate of 0.5 to prevent overfitting. The code for the VGG19 model was adapted from `https://github.com/machrisaa/tensorflow-vgg`.

4.3 Evaluation Metrics

We used several metrics to evaluate the performance of the proposed approaches on the test data. First, we evaluated the performance of the VGG19

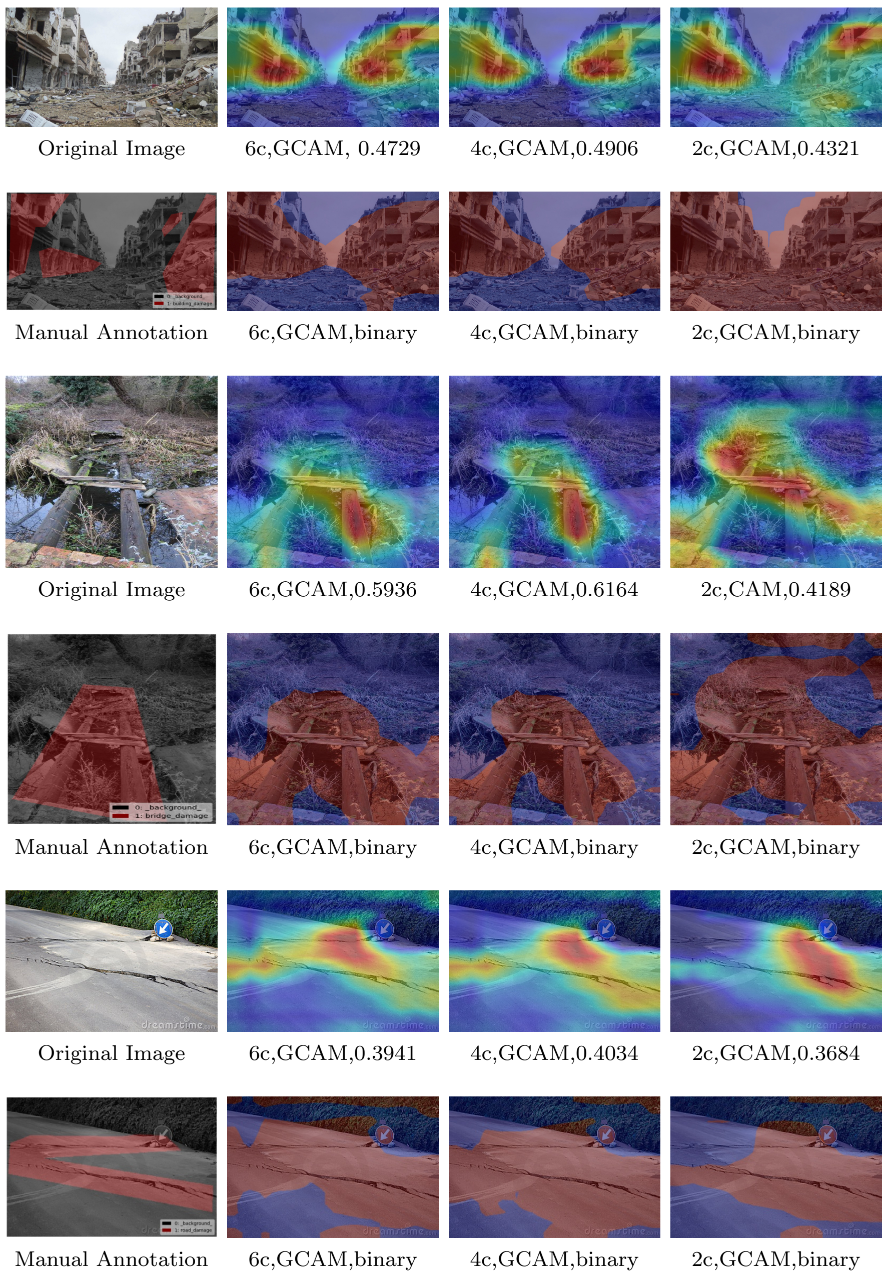


Fig. 4: Visualization of Damage Detection Maps for a building image (top two rows), a bridge image (middle two rows) and a road image (last two rows). The first row corresponding to an image shows the original image and the DDM maps produced with a six-class (6c), four-class (4c) and two-class (2c) VGG19 model, respectively. The Grad-CAM (GCAM) is used with all models. For each model, the IoU value of the image is also shown. The second row corresponding to an image shows the manual damage annotation of the image, together with the binary maps obtained from the DDM heatmaps (the binary maps are used to calculate the IoU values).

models using the classification accuracy, precision, recall and F1-measure (by comparing the predicted image labels with the ground truth labels). The ability of the Grad-CAM and Grad-CAM++ approaches to produce DDM heatmaps that accurately localize the damage was evaluated using the average intersection-over-union (IoU) metric [10] over all correctly classified images. The IoU metric is defined as follows:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{6}$$

where the 'Area of Overlap' and 'Area of Union' are computed with respect to a ground truth image, where damage is manually marked. IoU takes values in $[0, 1]$. The IoU value is large, when the predicted damage area is similar to the ground truth damage area.

To compare heatmaps with images where damage is manually marked in terms of IoU values, we transform the heatmaps to a binary representation as follows: we determine the maximum value in S_C and use 20% of the maximum value as a cutoff value for including a region in the disaster damage "object" or not [40]. In other words, only the regions in DDM with values larger than the cutoff value will be part of the localized damage. In the resulting transformed image (as well as in the human annotated images), the damage pixels have value 255, while no damage pixels have value 0.

To evaluate the ability of the proposed approaches to properly detect "damage areas/objects" we consider different IoU detection thresholds k (i.e., an object is detected if its IoU with the ground object is greater than the threshold k), and calculate the average precision over the detected objects. This metric is denoted as $AP@k$.

Finally, the ability of the DAV scores to capture the severity of the damage was evaluated using the root mean square error (RMSE).

5 Experimental Results and Discussion

In this section, we show the results of the experiments for evaluating the damage detection maps produced by our approach on the bridge. building and road categories, as well as the damage assessment value computed based on the damage detection maps. Finally, we use the results of the experiments to answer the research questions that we raised in Section 4.

5.1 Damage Detection Map Evaluation

The results of the experiments on the Google test dataset are shown in Table 3. The table first shows the accuracy of the VGG19 models used on the test data, as intuitively, a more accurate model will lead to a better DDM map. As can be seen in the table, the two-class VGG19 models are more accurate than the four-class and six-class models. The two-class model that discriminates between *road-damage* and *no-road-damage* is the most accurate among the two-class models, while the six-class model has the lowest accuracy overall. The same pattern is observed also for precision, where the two-class models perform better than the four-class and six-class models. In terms of recall, the two-class building and two-class road models have recall above 90%, while the two-class bridge model has recall around 75%, most probably due to the class imbalance in this dataset. Overall, the two-class road model is the best in terms of accuracy and F1-measure. The two-class road model is also the best in terms of the average IoU regardless of the CAM approach used to construct the DDM heatmap, but the six-class model is slightly better than the four-class model. Given the class imbalance, the two-class bridge model has worse IoU values as compared to the other models. We should also note that the standard deviation for all IoU values is relatively high, which means that some some images have high IoU, while others have low IoU. A sample image in each category, together with its manual annotation, the DDM heatmaps obtained with different models, and the corresponding binary maps are shown in Figure 4. As can be seen, the heatmaps produced by different models are not very different, but they lead to different binary maps and consequently different IoU values. Finally, it can be seen that overall the Grad-CAM approach leads to better average IoU values as compared to the Grad-CAM++ approach, regardless of the model used.

In addition to accuracy, precision, recall, F1-measure and average IoU, Table 3 also shows the average precision for each category of interest (i.e., bridge, building, and road) at different IoU thresholds k (specifically, $k = 0.5$, 0.4, and 0.3) with different VGG19 models (six-class, four-class and two-class) and different CAM approaches (Grad-CAM and Grad-CAM++). For a particular VGG19 model, one of the two CAM approaches is used to create a heatmap. Subsequently, a "damage object" is detected by taking a threshold on the values in the heatmap, as explained in Section 4.3. The IoU between the detected object and the corresponding ground truth object is computed. If the IoU is greater than the threshold k, the object is considered to be a true positive, otherwise it is considered to be a false positive. This information is used to compute $AP@k$. Intuitively, a higher threshold k may lead to some objects not being detected, while a lower threshold k may lead to false positives. As can be seen in Table 3, when using the six-class and four-class models, for the *bridge* category the $AP@0.4$ is worse than then $AP@0.5$ and also worse than $AP@0.3$, while for the other two categories, the values increase as k decreases. A similar pattern is consistently observed for the two-class models, where the performance increases as k decreases. It can also be seen that Grad-CAM generally gives better results than Grad-CAM++. When comparing the models for a particular category, we can see that the two-class road model is competitive in terms of average precision for the road category, although the four-class model perform well on this category as well. However, the six-class model is the best overall in terms of average precision for the bridge and building categories.

5.2 Damage Assessment Value Evaluation

To evaluate the ability of the DAV scores to capture damage severity, we used the two-class VGG19 model fine-tuned to discriminate between *building-damage* and *no-building-damage* and tested it on the *Building Damage Severity* dataset. More specifically, we first used the two-class VGG19 building model and Grad-CAM/Grad-CAM++ to create a re-scaled DDM heatmap S_C for each test image in the *Building Damage Severity* dataset. Subsequently, we used the heatmap S_C to compute a DAV score for each test image. Finally, we computed the RMSE over the set of images in the *Building Damage Severity* dataset, by comparing the DAV score of an image with the numeric annotation of the image (which captures damage severity). The resulting RMSE value for the Grad-CAM approach is 0.2538, while the RMSE value for the Grad-CAM++ approach is 0.4189, a result which further confirms the previous finding that Grad-CAM performs better than Grad-CAM++ in terms of damage detection and quantification.

The DAV values for the sample images in Fig. 3 are shown below the corresponding heatmap images in Fig. 5. As can be seen, images that present a more severe damage scene have higher DAV values, while images with less damage have smaller DAV scores. Thus, the DAV scores accurately reflect the degree of damage.

5.3 Discussion of the Results

Given the results presented above, in this subsection we answer the research questions that we raised in Section 4.

1. *Can the Damage Detection Map accurately locate the damage areas?* Conventionally, if the IoU value corresponding to a detected object (marked with a bounding box) is larger than 0.5, the detection/localization of that object is considered to be correct [10]. However, given that the damage is not an object but a concept, we find that the 0.5 threshold is too strict in the context of detecting and localizing damage. To explain this, we got an estimate for the average IoU agreement between two human annotators. Specifically, we had a second annotator mark damage for a random subset of the images in our dataset (precisely, 768 images). The average IoU between the two manual annotations was 0.5356, with standard deviation 0.2323 (larger than the standard deviation obtained with the automated approaches). Furthermore, recall that that the average precision for different categories was generally best for IoU threshold $k = 0.3$. This result, together with the relatively low agreement between annotators and the large standard deviation, suggests that the average IoU obtained using the automated approaches, which is as high as 0.4184 for road damage, is an indication that the damage detection map does a reasonable job at identifying infrastructure damage. This can also be seen from the samples images in Figure 6, which shows both images with DDMs that produce high IoU

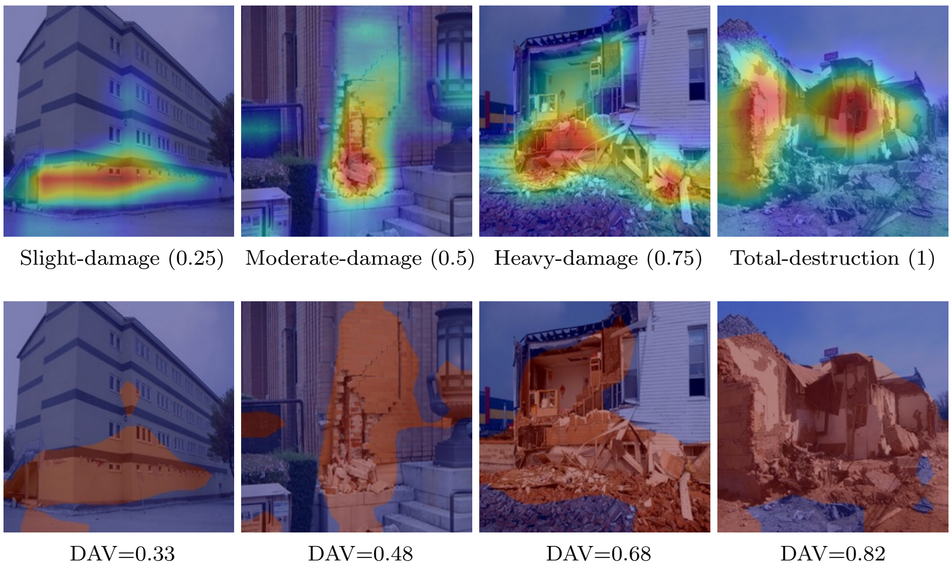


Fig. 5: Heatmaps (top row) and the corresponding binary maps (bottom row) for the sample images from the Building Damage Severity dataset, together with their severity annotations (shown below the top images) and the DAV scores (shown below the bottom images). The DAV scores properly reflect the severity of the damage.

values, and images with DDMs that produce low IoU values. As can be seen, for some of the images with low IoU values (e.g., the building image), one can argue that the annotations may not best capture the damage area in the first place, while for others (e.g., the bridge image), the model fails to annotate the most relevant damage area.

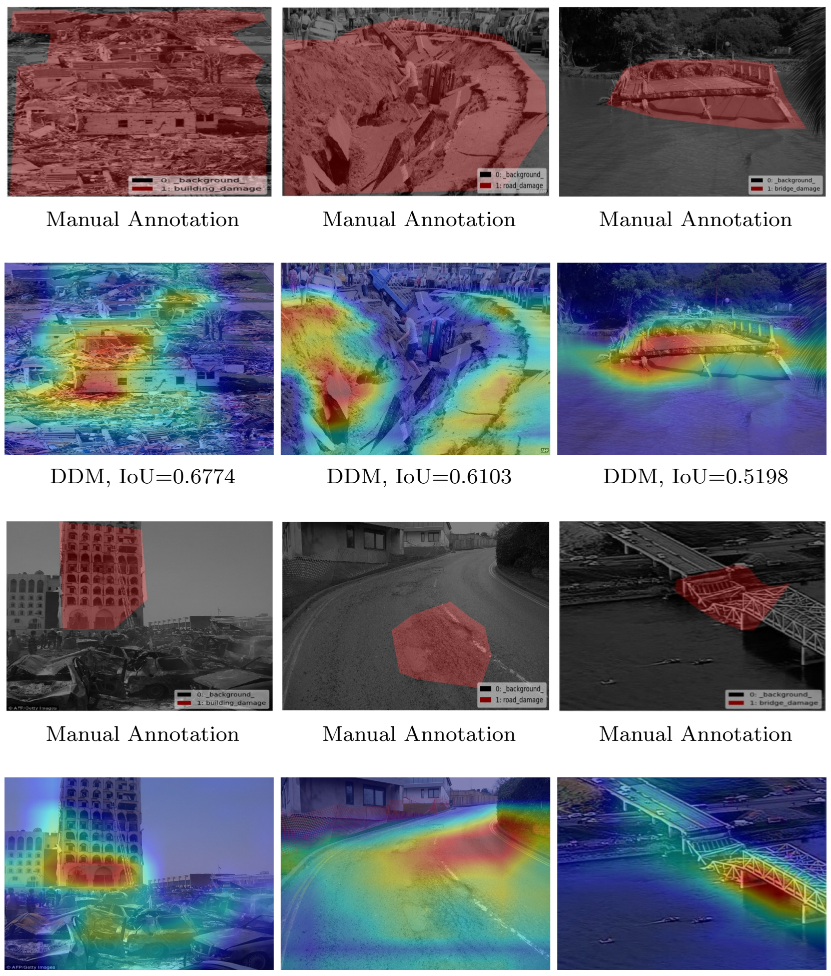
2. *What type of damage can be localized more accurately?* The two-class VGG19 road model is the most accurate and produces the highest average IoU values. Furthermore, a high average precision value (specifically, 0.722) is obtained for the road category when using the same two-class VGG19 model and an IoU threshold of 0.3. However, the highest average precision value overall is 0.7267, and it is obtained for the building category with the six-class VGG19 model. Thus, we can say that both road damage and building damage can be detected with high accuracy. For the bridge category, the highest average precision value is 0.5738 and it is also obtained with the six-class VGG model. One reason for bridge damage being harder to localize may come from the fact that bridge damage can be mistaken for road damage or building damage, as has been observed when analyzing the original labels of the images in the Google dataset.
3. *Can the DAV scores provide a reliable measure for damage severity?* Our results show that the RMSE between the DAV scores produced by our ap-

	six-class all	four-class all	two-class bridge	two-class building	two-class road
Accuracy	0.8099	0.8421	0.8143	0.9325	0.9550
Precision	0.8232	0.8489	0.9722	0.9184	0.9775
Recall	0.8099	0.8421	0.7447	0.9053	0.9158
F1-measure	0.8134	0.8441	0.8434	0.9173	0.9456
Avg. IoU (Grad-CAM)	0.3299	0.3132	0.2477	0.3361	0.4184
Std. IoU (Grad-CAM)	0.1758	0.1837	0.1575	0.1971	0.1745
Avg. IoU (Grad-CAM++)	0.3148	0.3054	0.2345	0.3366	0.3836
Std. IoU (Grad-CAM++)	0.1825	0.1876	0.1565	0.1926	0.1644
AP@0.5 bridge (Grad-CAM)	0.3167	0.2935	0.2803	N/A	N/A
AP@0.5 building (Grad-CAM)	0.5221	0.3493	N/A	0.4424	N/A
AP@0.5 road (Grad-CAM)	0.2454	0.3049	N/A	N/A	0.2718
AP@0.5 bridge (Grad-CAM++)	0.3167	0.2934	0.2038	N/A	N/A
AP@0.5 building (Grad-CAM++)	0.5156	0.3560	N/A	0.4559	N/A
AP@0.5 road (Grad-CAM++)	0.2416	0.2516	N/A	N/A	0.2725
AP@0.4 bridge (Grad-CAM)	0.5738	0.5526	0.2006	N/A	N/A
AP@0.4 building (Grad-CAM)	0.5425	0.5159	N/A	0.4729	N/A
AP@0.4 road (Grad-CAM)	0.4172	0.4678	N/A	N/A	0.4547
AP@0.4 bridge (Grad-CAM++)	0.2349	0.2934	0.2146	N/A	N/A
AP@0.4 building (Grad-CAM++)	0.5420	0.5160	N/A	0.4943	N/A
AP@0.4 road (Grad-CAM++)	0.3749	0.4757	N/A	N/A	0.3952
AP@0.3 bridge (Grad-CAM)	0.4537	0.4270	0.5005	N/A	N/A
AP@0.3 building (Grad-CAM)	0.7267	0.7098	N/A	0.6049	N/A
AP@0.3 road (Grad-CAM)	0.5881	0.6998	N/A	N/A	0.7222
AP@0.3 bridge (Grad-CAM++)	0.4537	0.4270	0.4455	N/A	N/A
AP@0.3 building (Grad-CAM++)	0.7178	0.6639	N/A	0.6113	N/A
AP@0.3 road (Grad-CAM++)	0.5589	0.6558	N/A	N/A	0.6918

Table 3: Experimental results on the Google test dataset

proach and the ground truth damage severity annotations is 0.2538, when the Grad-CAM approach is used to generate the DDM heatmaps and subsequent DAV scores. Given that the damage severity is in the range [0,1], we find this result to be very promising, in other words, we can claim that the DAV score provide a good measure for damage severity, although the reliability can be further improved. While the dataset that we used in this experiment is relatively low, it is the only dataset we found publicly available, which is labeled with respect to damage severity. However, given the transferability power of CNNs, and the fact that we used a pre-traied VGG network, our model may not improve significantly with a larger number of training images. To validate this claim, we plan to collaborate with damage infrastructure experts in the future to produce a larger damage severity dataset.

4. *Among two-class, four-class, or six-class models, what model leads to better results overall?* Overall, the six-class VGG19 model together with Grad-CAM works best for our set of images, although the four-class model is also competitive, and the two-class VGG19 road model is the best for road damage. Intuitively, the six-class model, which is trained to discriminate between *bridge-damage*, *no-bridge-damage*, *building-damage*, *no-building-*



Manual Annotation | Manual Annotation | Manual Annotation

DDM, IoU=0.6774 | DDM, IoU=0.6103 | DDM, IoU=0.5198

Manual Annotation | Manual Annotation | Manual Annotation

DDM, IoU=0.1732 | DDM, IoU=0.1663 | DDM, IoU=0.1890

Fig. 6: Examples of images with DDM heatmaps that lead to high/low IoU values. The first two rows show images with high IoU values, while the last two rows show images with low IoU values.

damage, *road-damage*, and *no-road-damage*, learns not only features that are indicative of damage versus no-damage, but also features that are indicative of a specific category among the three target categories (bridge, building, road). As opposed to that, the four-class model, which is train to discriminate between *bridge-damage*, *building-damage*, *road-damage*, and *no-damage* may learn features that identify no-damage in general, as opposed to no-damage in a particular category (in addition to features for discriminating between different categories), and thus leads to worse results than the six-class models. While the two-class road model works the best for road damage, as expected, it seems surprising that the two-class bridge and building models are generally worse than the other models. Given that some bridge images look similar to building images, it may be that the richer features extracted from the combined dataset help get better DDM heatmaps. As opposed to that, road damage might be more distinctive and easier to identify with a two-class model.

5. *Between Grad-CAM and Grad-CAM++, which approach is better?* Our results show that the Grad-CAM approach is consistently better than the Grad-CAM++ approach. This may seem surprising, as Grad-CAM++ is an extension of Grad-CAM, which has been shown to give better results on datasets with multiple occurrences of an object in an image. However, in the context of damage localization, damage is rather as a concept with soft boundaries rather than an object with hard boundaries, and as such, it doesn't present multiple occurrences. The Grad-CAM++ may highlight additional regions that are not representative of damage, as marked by human annotators. Therefore, Grad-CAM++ can lead to worse results as compared to Grad-CAM, which focuses on the most representative regions for the damage.

6 Conclusion

Given the large number of social media images posted by eyewitnesses of disasters, we proposed an approach for detecting and localizing disaster damage at low cost. Our approach is built on top of a fine-tuned VGG19 model, and utilizes the Grad-CAM approach to produce a DDM heatmap. Furthermore, the DDM is used to calculate a DAV score for each image. This scoring is performed on a continuous scale and can be used to assess the severity of the damage. The DAV score, together with the DDM heatmap, can be used to identify and prioritize useful information for disaster response, while providing visual explanations for the suggestions made to increase the trust in the computational models. Quantitative and qualitative evaluations of DDM and DAV components show the feasibility of our proposed approach.

As part of future work, it is of interest to study the applicability of the proposed approach to estimate the global damage produced by a disaster based on aggregating the DAV values from individual images. Also, geo-tagging images would enable disaster response teams not only to identify damage, but also to find its physical location.

7 Acknowledgements

The computing for this project was performed using Amazon Web Services (AWS). We thank the National Science Foundation and Amazon Web Services for support from grant IIS-1741345, which enabled the research and the computation in this study. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either express or implied, of the National Science Foundation or AWS.

References

1. Alam, F., Imran, M., Ofli, F.: Image4act: Online social media image processing for disaster response. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 601–604. ACM (2017)
2. Ashktorab, Z., Brown, C., Nandi, M., Culotta, A.: Tweedr: Mining twitter to inform disaster response. Proc. of ISCRAM (2014)
3. Attari, N., Ofli, F., Awad, M., Lucas, J., Chawla, S.: Nazr-cnn: Fine-grained classification of uav imagery for damage assessment. In: Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on, pp. 50–59. IEEE (2017)
4. Bica, M., Palen, L., Bopp, C.: Visual representations of disaster. In: Proc. of the 2017 ACM CSCW, pp. 1262–1276. ACM, NY, USA (2017)
5. Caragea, C., Silvescu, A., Tapia, A.H.: Identifying informative messages in disasters using convolutional neural networks. In: Proc. of the ISCRAM, Brazil (2016)
6. Castillo, C.: Big Crisis Data: Social Media in Disasters and Time-Critical Situations. Cambridge University Press (2016)
7. Cha, Y.J., Choi, W., Büyüköztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering **32**(5), 361–378 (2017)
8. Chattopadhyay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. CoRR **abs/1710.11063** (2017)
9. Enenkel, M., Saenz, S.M., Dookie, D.S., Braman, L., Obradovich, N., Kryvasheyeu, Y.: Social media data analysis and feedback for advanced disaster risk management. CoRR **abs/1802.02631** (2018)
10. Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes (voc) challenge. Int. J. Comput. Vision **88**(2), 303–338 (2010)
11. Fan, Y., Wen, Q., Wang, W., Wang, P., Li, L., Zhang, P.: Quantifying disaster physical damage using remote sensing dataa technical work flow and case study of the 2014 ludian earthquake in china. International Journal of Disaster Risk Science **8**(4), 471–488 (2017)
12. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)
13. Guan, X., Chen, C.: Using social media data to understand and assess disasters. Natural hazards **74**(2), 837–850 (2014)
14. Gueguen, L., Hamid, R.: Large-scale damage detection using satellite imagery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1321–1328 (2015)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
16. Huang, Q., Xiao, Y.: Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery. ISPRS Int. Journal of Geo-Information **4**(3) (2015)
17. Imran, M., Castillo, C., Diaz, F., Vieweg, S.: Processing social media messages in mass emergency: A survey. ACM Computing Surveys (CSUR) **47**(4), 67 (2015)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
19. Kryvasheyeu, Y., Chen, H., Obradovich, N., Moro, E., Van Hentenryck, P., Fowler, J., Cebrian, M.: Rapid assessment of disaster damage using social media activity. Science advances **2**(3) (2016)

20. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
21. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4) (1989)
22. Li, H., Caragea, D., Caragea, C., Herndon, N.: Disaster response aided by tweet classification with a domain adaptation approach. Journal of Contingencies and Crisis Management (JCCM), Special Issue on HCI in Critical Systems. **26**(1), 16–27 (2017)
23. Li, X., Zhang, H., Caragea, D., Imran, M.: Localizing and quantifying damage in social media images. In: IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining (ASONAM), pp. 194–201 (2018)
24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: In European conference on computer vision, p. 2137. Springer (2016)
25. Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H.: Road damage detection using deep neural networks with images captured through a smartphone. arXiv preprint arXiv:1801.09454 (2018)
26. Meier, P.: Digital Humanitarians: How Big Data Is Changing the Face of Humanitarian Response. CRC Press, Inc., FL, USA (2015)
27. Nguyen, D.T., Al-Mannai, K., Joty, S.R., Sajjad, H., Imran, M., Mitra, P.: Rapid classification of crisis-related data on social networks using convolutional neural networks. CoRR **abs/1608.03902** (2016)
28. Nguyen, D.T., Ofli, F., Imran, M., Mitra, P.: Damage assessment from social media imagery data during disasters. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 569–576. ACM (2017)
29. Nia, K.R., Mori, G.: Building damage assessment using deep learning and ground-level image data. In: 14th Conference on Computer and Robot Vision (CRV), pp. 95–102. IEEE (2017)
30. Palen, L., Anderson, K.M.: Crisis informatics-new data for extraordinary times. Science **353**(6296), 224–225 (2016)
31. Resch, B., Usländer, F., Havas, C.: Combining machine-learning topic models and spatiotemporal analysis of social media data for disaster footprint and damage assessment. Cartography and Geographic Information Science pp. 1–15 (2017)
32. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision (ICCV) (2017)
33. Sen, A., Rudra, K., Ghosh, S.: Extracting situational awareness from microblogs during disaster events. In: 7th Int. Conf. on Communication Systems and Networks (COMSNETS), pp. 1–6. IEEE (2015)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
35. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR) (2015)
36. Vetrivel, A., Gerke, M., Kerle, N., Nex, F., Vosselman, G.: Disaster damage detection through synergistic use of deep learning and 3d point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. ISPRS journal of photogrammetry and remote sensing (2017)
37. Xie, S., Duan, J., Liu, S., Dai, Q., Liu, W., Ma, Y., Guo, R., Ma, C.: Crowdsourcing rapid assessment of collapsed buildings early after the earthquake based on aerial remote sensing image: A case study of yushu earthquake. Remote Sensing **8**(9), 759 (2016)
38. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems, pp. 3320–3328 (2014)
39. Yuan, F., Liu, R.: Feasibility study of using crowdsourcing to identify critical affected areas for rapid damage assessment: Hurricane matthew case study. International Journal of Disaster Risk Reduction (2018)

40. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2929 (2016)