Identifying Disaster Damage Images Using a Domain Adaptation Approach

ABSTRACT

Approaches for effectively filtering useful situational awareness information posted by eyewitnesses of disasters, in real time, are greatly needed. While many studies have focused on filtering textual information, the research on filtering disaster images is more limited. In particular, there are no studies on the applicability of domain adaptation to filter images from an emergent target disaster, when no labeled data is available for the target disaster. To fill in this gap, we propose to apply a domain adaptation approach, called domain adversarial neural networks (DANN), to the task of identifying images that show damage. The DANN approach has VGG-19 as its backbone, and uses the adversarial training to find a transformation that makes the source and target data indistinguishable. Experimental results on several pairs of disasters suggest that the DANN model generally gives similar or better results as compared to the VGG-19 model fine-tuned on the source labeled data.

Keywords

Image classification, disaster damage, domain adaptation, domain adversarial neural networks.

INTRODUCTION

The increased popularity of social media websites has transformed the way in which affected populations communicate with response organizations during and following major disasters. Victims of several recent hurricanes, including Hurricane Harvey, Hurricane Irma and Hurricane Florence, have turned to social media to request help and assistance, as emergency hotlines were sometimes unreachable due to the high volume of calls (Rhodan 2017; MacMillan 2017; Frej 2018). According to Rhodan (2017), a woman and two children were rescued during Hurricane Harvey after the woman posted a desperate message for help on Twitter: 'I have 2 children with me and tge [sic] water is swallowing us up. Please send help.' Similarly, an image posted on Twitter, which showed 'elderly residents sitting in greenish flood water,' raised awareness of a nursing home situation and resulted in urgent help being sent to the nursing home (Rhodan 2017).

While social media can help increase situational awareness, inform rescue operations, and save lives, its value is highly unexploited by response teams, in part due to the lack of tools that can help filter actionable information from the big data posted during disasters. A recent study (Villegas et al. 2018) reported that during Hurricane Harvey, FEMA missed 46% of the critical damage information posted by affected individuals on Twitter, and thus many areas heavily impacted by the hurricane were missed from the original damage estimates provided by FEMA. Similarly, Kryvasheyeu et al. (2016) used Hurricane Sandy tweets, and Enenkel et al. (2018) used Hurricanes Harvey and Irma filtered, geo-located tweets to show that rapid early damage assessment can be facilitated by social media. Among others, the above-mentioned studies (Kryvasheyeu et al. 2016; Villegas et al. 2018; Enenkel et al. 2018) suggest that tools that can identify critical social media information in real-time are greatly needed, and should be incorporated into the operational decision-making pipelines of response organizations.

To meet these requirements, many studies have focused on the design of machine learning tools to identify relevant, informative, actionable situational awareness information in social media (Vieweg et al. 2010; Sen et al. 2015; Huang and Xiao 2015; Imran et al. 2016). More recently, deep learning approaches have also been proposed in the context of identifying information useful for disaster response on social media (Caragea et al. 2016; Nguyen, Al-Mannai, et al. 2017; Neppalli et al. 2018; Derczynski et al. 2018; Aipe et al. 2018), and have been generally shown to produce better results than the traditional supervised learning approaches. As opposed to research on classifying disaster tweets, research on classifying and retrieving useful information from disaster images is still in its early stage, although recent studies have suggested that useful social awareness information can be found in social media images (Bica et al. 2017; Lagerstrom et al. 2016; Alam, Ofli, et al. 2018b). To advance the state-of-the-art in

this area, Alam, Ofli, et al. (2018) and Mouzannar et al. (2018) recently published multi-modal datasets, consisting of both tweet text and images. Furthermore, Mouzannar et al. (2018) developed a deep learning approach to identify damage images in their dataset. Related to this, Nguyen, Ofli, et al. (2017) proposed to classify disaster images according to damage severity using fine-tuned convolutional neural networks, and X. Li et al. (2018) proposed a method based on class activation mapping (CAM) to localize and quantify damage in social media images posted during disasters.

While significant contributions have been made in terms of developing traditional and deep learning models to identify useful situational awareness information in social media, most of the existing approaches are supervised learning approaches, which require labeled data to train accurate models. It is unrealistic to assume that labeled data is readily available in the early hours of a disasters, when help may be most needed. Domain adaptation or transfer learning approaches (Pan and Yang 2010), which make use of labeled data from a prior disaster (called *source*) and unlabeled data from the emergent disaster (called *target*), have been shown to lead to better results as compared to supervised classifiers learned from a prior disaster, when used for target tweet classification (H. Li et al. 2017; Mazloom et al. 2018). Domain adaptation has been used extensively for image classification using deep neural networks (Wang and Deng 2018), including social media image classification by fine-tuning a pre-trained deep neural network, such as VGG-16 or VGG-19 (Simonyan and Zisserman 2014), as in (Nguyen, Ofli, et al. 2017; X. Li et al. 2018). However, domain adaptation from a source disaster to a target disaster has not been explored in the context of classifying social media images posted during and shortly after a disaster. Thus, the goal of this study is to gain insights into the usefulness of transferring knowledge from a source disaster to a target disaster, when classifying images according to damage severity.

We propose to use an approach called Domain-Adversarial Neural Network (DANN) introduced by Ganin et al. (2016). The DANN approach is based on domain-adversarial training, which reduces the shift between the source and target domains, by making the source and target feature representations indiscriminate, while the source feature representation is discriminative for the source classification task. Experimental results on images from several source-target disaster pairs have shown that the DANN approach, which uses source labeled data and unlabeled target data, can improve the classification accuracy obtained with a deep neural network trained only from source labeled data, especially if the two disasters are of different type (e.g., a hurricane and an earthquake).

The contributions of the paper are as follows:

- We have adapted the DANN model by combining it with VGG-19 to benefit from the VGG-19 extensive training, as shown in the "Background and Approach" section.
- We have performed extensive experiments on several pairs of disasters, and studied the ability of the adapted model to transfer information about image damage from a source disaster to a target disaster. The experimental setup and results are described in "Experimental Setup" and "Experimental Results" sections, respectively.
- We have performed visualization of the original and transformed representations of the source versus target images to gain insights into the results of the model, as shown in the "Visual Distribution Analysis" section.

BACKGROUND AND APPROACH

In this section, we introduce some background on image classification, and describe the approach used to perform domain adaptation from a source disaster to a target disaster. Figure 1 shows an overview of the model used, which has the VGG-19 network (Simonyan and Zisserman 2014) as its backbone, and uses the domain-adversarial training (Ganin et al. 2016) to reduce the domain shift.

Adapted VGG-19 Architecture

Convolutional neural networks (CNN) have achieved impressive results in image analysis and computer vision. Many network architectures have been proposed for image classification in recent years, including AlexNet (Krizhevsky et al. 2012), VGG-19 (Simonyan and Zisserman 2014), ResNet (He et al. 2016). ResNet and VGG-19 models have won the ImageNet competition[1], which proves that these models have good performance on image classification. We choose VGG-19 as the backbone for our model given that this network is relatively simpler than ResNet.

As can be seen in Figure 1, VGG-19 consists of 16 convolutional layers and 3 fully connected layers. A convolutional layer can be seen as a feature extractor, as it extracts image fragments (e.g., edge) that are important with respect to

[1] http://image-net.org/challenges/LSVRC/

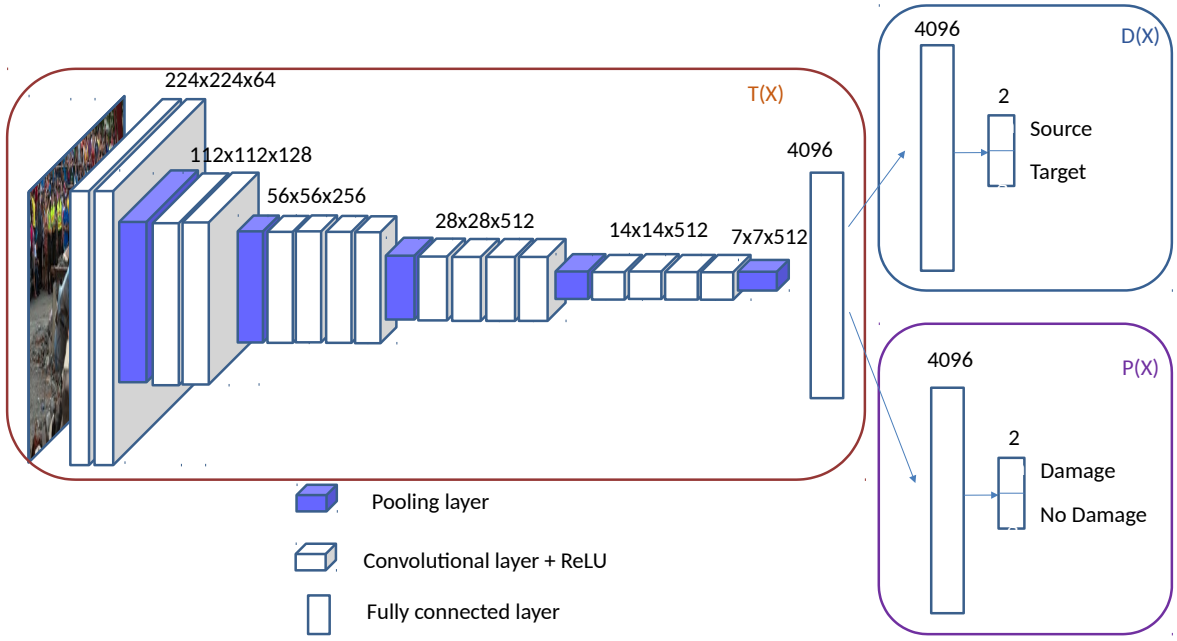


Figure 1. Overview of the Model Architecture.

the classification task, while ignoring noisy fragments which do not contribute to classification. Each convolutional layer is accompanied by a non-linear ReLU activation. The 5 max pooling layers shown in the figure are used to achieve dimensionality reduction and further filter out noise. The standard VGG-19 has two fully connected layers with dimension 4096, and one softmax layer with dimension 1000. The softmax layer has dimension 1000 because VGG-19 was originally trained for a 1000-class object classification task. However, we aim to adapt VGG-19 to classify disaster images in two classes: *damage* or *no-damage*, and thus, we have changed the dimension of the softmax layer from 1000 dimensions to 2 dimensions. The VGG-19 model includes more then 130 million parameters that need to be learned. It is unfeasible to learn all those parameters from scratch. Instead, we have used the pre-trained VGG-19 model to initialize the parameter values in our network for all, but the last fully connected layer (as that layer has a different dimension), and subsequently fine-tuned all parameters of the model based on our disaster data. Given that the pre-trained VGG-19 model is learned from millions of images, it is known to generalize well to other sets of images in terms of informative features that it extracts. Thus, we can fine-tune it to our specific classification task with a relatively small number of images.

Domain-Adversarial Neural Networks

Domain Adaptation is a machine learning setting, where data from a source domain, X_s, is used to predict a target domain, X_t, under the assumption that the source and target domains have different distributions (a.k.a., domain shift), but share some similar patterns. In unsupervised domain adaptation, the labels of the source domain data, denoted as y_s, are available, while the labels of the target domain data, y_t, are not available. The task is to learn a classifier for the target data, using the labeled source data and the unlabeled target data. A standard approach to domain adaptation is to transform the source and target data representation (a.k.a., feature adaptation), so that that the source and target distributions become indistinguishable (Pan and Yang 2010). More precisely, one needs to identify a transfer function, $T(X)$, to transform the original source, X_s, and target, X_t, domain data into $T(X_s)$ and $T(X_t)$ data which have similar distributions. Intuitively, a classifier learned on $T(X_s)$ with labels y_s should presumably have a good predictive performance on $T(X_t)$. Thus, the domain adaptation problem is reduced to finding the transfer function $T(X)$. Inspired by the generative adversarial network (GAN) proposed by Goodfellow et al. (2014), Ganin et al. (2016) designed a Domain-Adversarial Neural Network (DANN), which includes a component that explicitly aims to reduce the shift between a source and a target.

We have adapted the DANN model proposed by Ganin et al. (2016) by combining its adversarial training with the VGG-19 network, to take advantage of the extensive training of VGG-19. The architecture of the adapted DANN network, is shown in Figure 1. As can be seen in Figure 1, our model includes three components: 1) a transfer network, denoted by T, which enables the transfer of information from the source to the target, by reducing the shift

between the two domains (in our case, between the two disasters); 2) a domain discriminative network, denoted by D, which identifies if an input image is from the source or from the target domain; 3) a prediction network, denoted by P, which is used to predict if an image is in the *damage* category or in the *no-damage* category. The input for the transfer network, T, consists of images from either source or target domain. After applying the transformation T to the source or target images, the transformed images are classified as *source* or *target* by the discriminative network D, and classified as *damage* or *no-damage* by the predictor network P. Informally, the idea of the adversarial training is to train a good domain discriminant to separate the source and target domains, and at the same time, train a transfer network to make the two domain look so similar, that even a good discriminant network can not separate them. The predictor network, P, is trained on the transformed source data and then used to classify the transformed target data.

As can be seen in Figure 1, the the transfer network, $T(X)$, consists of the first 16 convolutional/pooling layers of VGG-19, together with the first fully connected layer in VGG-19. The domain discriminative network, $D(X)$, and prediction network, $P(X)$, have similar architectures, which are equivalent to the last two fully connected layers of VGG-19. Specifically, $D(X)$ has one fully connected layer with dimension 4096, and one fully connected softmax layer with two dimensions, corresponding to the source (S) and the target (T) domains, respectively. Similarly, $P(X)$ has one fully connected layer with dimension 4096, and one fully connected softmax layer with two dimensions, corresponding to *damage* and *no-damage* categories, respectively.

DANN Model Training

The DANN is trained by minimizing the total loss of the model shown in Figure 1. To define the total loss function, we first define the following cross-entropy loss functions, corresponding to the three sub-networks in our model:

$\mathcal{L}_{prediction} = \min_{P} \mathcal{L}(y_s, P(T(x)))$, where y_s is the true class label of the transform instance x.

$\mathcal{L}_{domain} = \min_{D} \mathcal{L}(TDL, D(T(x)))$, where TDL is the true domain label, i.e., S for *source*, and T for *target*.

$\mathcal{L}_{transfer} = \min_{T} \mathcal{L}(FDL, D(T(x)))$, where FDL is a fake domain label, i.e., S for *target*, and T for *source*.

The total loss is defined as follows: $\mathcal{L}_{total} = \mathcal{L}_{domain} + \mathcal{L}_{transfer} + \mathcal{L}_{prediction}$

Let θ_D denote all the parameters in $D(X)$, θ_T denote all the parameters in $T(X)$, and θ_P denote all the parameters in $P(X)$. At each training step (a pass through the source data), we first compute each loss, $\mathcal{L}_{domain}$, $\mathcal{L}_{transfer}$, and $\mathcal{L}_{prediction}$, and update the parameters accordingly. However, we notice that the discriminant loss function always 'defeats' the transfer loss function. To account for this, we compute the transfer loss function two times in each step, as suggested in (Bang and Shim 2018). Then, the transfer and domain losses become almost equal, which means the transfer function reduced the difference between the two domains to an extent that the discriminant function can not separate the domains. The update rules are shown below:

$$\theta_D \longleftarrow \theta_D - \mu \frac{\partial \mathcal{L}_{domain}}{\partial \theta_D} \quad (1)$$

$$\theta_T \longleftarrow \theta_T - \mu \frac{\partial \mathcal{L}_{transfer}}{\partial \theta_T} \quad (2)$$

$$\theta_T \longleftarrow \theta_T - \mu \frac{\partial \mathcal{L}_{prediction}}{\partial \theta_T} \quad (3)$$

$$\theta_P \longleftarrow \theta_P - \mu \frac{\partial \mathcal{L}_{prediction}}{\partial \theta_P} \quad (4)$$

where μ is the learning rate. We use the stochastic gradient descent (SGD) to optimize the parameters. The pseudocode for training the DANN network using SGD is shown in Algorithm 1. Parameter updates are performed for each batch sampled from source. Corresponding to a batch from source of size m (line 5 in the algorithm), which is used to train the prediction network, the algorithm also creates a batch that has half (i.e., $m/2$) source instances and half target instances (line 6 in the algorithm). The source/target batch is used to train the transfer and domain networks.

EXPERIMENTAL SETUP

In this section, we present the experimental setup, including the dataset used, cross-validation setup, baselines, and implementation details. The experiments conducted in this study were designed to answer the following research questions: (1) How does the DANN model work for adaptation between disasters of the same type (e.g., two earthquakes) versus adaptation between disasters of different types (e.g., an earthquake and a hurricane)? (2) Does the source-target feature adaptation through DANN give better results as compared to the direct source adaptation from VGG-19? (3) How do the results of the DANN model compare to the results of a VGG-19 network adapted based on data from the target domain itself?

Algorithm 1 DANN Training Using SGD

Input: Datasets $S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ (where n_s represents the number of instances in the source S) and $T = \{x_i^t\}_{i=1}^{n_t}$ (where n_t represents the number of instances in the target T)
Output: $\theta_T, \theta_P, \theta_D$

$\theta_T \leftarrow$ pre-trained VGG-19 weights
$\theta_P, \theta_D \leftarrow$ random initial weights
$Step\ k = 0$
for k from 1 to $steps$ **do**
  for each source batch $\{(x_i^s, y_i^s)\}_{i=1}^{m}$ **do**
    Sample source/target batch $(x_j^s, x_j^t)_{j=1}^{m/2}$
    $TDL_j = 1$ if $x_j \in$ source, 0 otherwise
    $FDL_j = 0$ if $x_j \in$ source, 1 otherwise
    Compute the loss $\mathcal{L}_{domain}$
    Update θ_D using Equation (1)
    Compute the loss $\mathcal{L}_{transfer}$
    Update θ_T using Equation (2)
    Compute the loss $\mathcal{L}_{transfer}$
    Update θ_T using Equation (2)
    Compute the loss $\mathcal{L}_{prediction}$
    Update θ_T and θ_P using Equations (3) and (4)

Table 1. Disaster class distribution, together with the total number of labeled images in each disaster.

Class	Nepal Earthquake	Ecuador Earthquake	Ruby Typhoon	Matthew Hurricane
Damage	11,183	933	433	204
No-damage	7,919	791	400	127
Total	19,102	1,724	833	331

Dataset

We used the disaster image dataset published by Nguyen, Alam, et al. (2017) in our experiments. The dataset contains images from four disasters, specifically Nepal Earthquake, Ecuador Earthquake, Ruby Typhoon, and Matthew Hurricane. In the original dataset, there are three damage classes: *severe*, *mild*, and *none*. However, Nguyen, Alam, et al. (2017) suggested that the task of discriminating between *mild* and *severe* damage is very subjective, and there is significant overlap in the dataset between the two classes. Therefore, we combine the classes *severe* and *mild* into one class called *damage*. Our goal is to identify images that include damage and separate them from images that do not show any damage. Table 1 shows the total number of images in each disaster, and also the class distribution for each disaster.

Setup

Given the four disasters in our dataset, we experiment with all possible source/target pairs. Thus, some source/target pairs consist of disasters of the same type (e.g., two earthquakes), while other pairs consist of disasters of different types (e.g., an earthquake and a hurricane). Furthermore, some pairs have smaller amounts of source data, while others have larger amounts of source data. Similarly, some pairs have a smaller amount of target unlabeled data, while other pairs have a larger amount of target unlabeled data. This variety of pairs makes it possible to answer the research questions we have raised.

For each source/target experiment, we randomly split the target data into target unlabeled (80%) and target test (20%). Each DANN model was trained on all source data and the target unlabeled data, and subsequently tested on the test dataset. The results are evaluated using four standard metrics, specifically, accuracy, precision, recall and F1-measure. For cross-validation purposes, we created three different random splits for each source/target pair. The metrics reported represent averages over the three random splits (together with the standard deviation).

Baselines

We compared the DANN model against two baselines: 1) a model that uses only the labeled source data to adapt the pre-trained VGG-19 – this model can generally be seen as a lower bound for domain adaptation; 2) a model that uses the target data as labeled data to adapt the pre-trained VGG-19 – this model can generally be seen as an upper bound for domain adaptation.

Implementation Details

We used TensorFlow's MomentumOptimizer procedure to train the model using the mini-batch stochastic gradient descent on a K80 graphics card. Based on our preliminary experimentation, we set the learning rate to 0.001, and the batch size to 64 images. Furthermore, we used the dropout technique with a rate of 0.5 to prevent overfitting. The code for the VGG-19 model was adapted from `https://github.com/machrisaa/tensorflow-vgg`.

EXPERIMENTAL RESULTS

The results of the experiments are shown in Table 2 for the DANN model, which was trained on source labeled and target unlabeled data (denoted as DANN-SL-TU), and for the two baselines, VGG-19-SL (VGG-19 adapted using source labeled data), and VGG-19-TL (VGG-19 adapted using target labeled data). The DANN model and the baselines are tested on the same target test data. Each experiment was repeated three times with different target unlabeled/test splits, and the results averaged over the three runs.

To answer our first research question, *How does the DANN model work for adaptation between disasters of the same type versus adaptation between disasters of different types?*, we compare the results of DANN-SL-TU with the results of VGG-19-SL for pairs of similar and different disasters. We consider Matthew Hurricane and Ruby Typhoon to be disasters of the same type, and the same for Ecuador Earthquake and Nepal Earthquake. Disaster pairs consisting of a hurricane/typhoon and an earthquake are considered to be different. As can be seen from Table 2, the results of DANN-SL-TU are similar and sometimes better than the results of VGG-19-SL, for pairs of similar disasters. Furthermore, the DANN-SL-TU results are generally better than the VGG-19-SL results for different disasters. For example, when we use either Ecuador Earthquake or Nepal Earthquake as source, the results for predicting the other earthquake are similar between DANN-SL-TU and VGG-19-SL. However, if we use either disaster as source, and predict Matthew Hurricane or Ruby Typhoon, DANN-SL-TU gives significantly better results. Similarly, when Ruby Typhoon is used as source, the results obtained with DANN-SL-TU for Nepal Earthquake and Ecuador Earthquake are significantly better then the results obtained with VGG-19-SL, while the two models are similar for Matthew Hurricane.

Given the above-mentioned results, the answer to our second question, *Does the source-target feature adaptation through DANN give better results as compared to the direct source adaptation from VGG-19?*, is overall positive, as the DANN-SL-TU model gives similar or better results than VGG-19-SL. However, if we consider the size of the datasets used in our experiments, the results in Table 2 suggest that the DANN-SL-TU approach gives better results if the source labeled data and the target unlabeled data used for training are relatively large (e.g., a few thousands). Furthermore, it can be seen that DANN increased the recall for the damage class in most case, especially when the source and target disasters are of different types. Finally, when comparing DANN-SL-TU with VGG-19-TL to answer our third question, *How do the results of the DANN model compare to the results of a VGG-19 network adapted based on data from the target domain itself?*, we observe that indeed VGG-19-TL acts as an upper bound for the DANN-SL-TU model. While the gap between the two models is significant, the gap between VGG-19-TL and VGG-19-SL is also large, suggesting that the source data distribution is indeed different from the target data distribution, and better domain adaptation models have the potential to bridge the gap.

Figure 2 shows examples of images that are correctly classified or miss-classified by the DANN-SL-TU/VGG-19-SL/VGG-19-TL networks, when Ruby Typhoon is used as target disaster and Ecuador Earthquake is used as source disasters. Specifically, the top row in Figure 2 shows examples of images that are correctly classified as *damage* by DANN-SL-TU and VGG-19-TL, but miss-classified as *no-damage* by VGG-19-SL. As the VGG-19-SL network is trained to recognize earthquake damage, it can fail to recognize rain/flood/wind damage present in typhoon images. As opposed to that, the middle row in Figure 2 shows images that are miss-classified as *damage* by both DANN-SL-TU and VGG-19-TL, but correctly classified as *no-damage* by VGG-19-SL. It seems that the networks that use images from the target typhoon during training may over-learn that water-related or wind-like images are showing damage. Finally, the bottom row Figure 2 shows images that are miss-classified by both DANN-SL-TU and VGG-19-SL. The first two images in the bottom row are correctly classified by VGG-19-TL as *damage* and *no-damage*, respectively, while the last two images are also miss-classified by VGG-19-TL as *damage*. Together

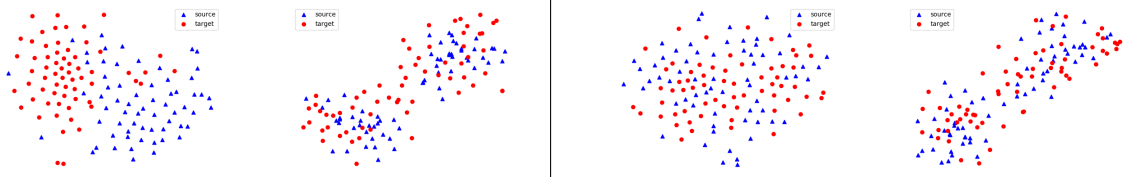
Table 2. Classification results: average accuracy (Acc.), precision (Prec.), recall (Rec.) and F1-measure (F1) for the damage class (average is taken over three runs), together with the corresponding standard deviation (in parentheses), for DANN-SL-TU (which uses source labeled data and target unlabeled data), VGG-19-SL (adapted using source labeled data), and VGG-19-TL (adapted using target labeled data). Highlighted in bold font are results that are statistically different when comparing VGG-19-SL and DANN-SL-TU using a paired t-test with $p \leq 0.05$.

Source	Ecuador Earthquake											
Target	Matthew Hurricane				Nepal Earthquake				Ruby Typhoon			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
VGG-19-SL	0.557	0.786	0.382	0.514	0.828	0.861	0.841	0.851	0.625	0.823	0.360	0.499
	(0.023)	(0.058)	(0.014)	(0.017)	(0.006))	(0.004)	(0.012)	(0.006)	(0.023)	(0.074)	(0.048)	(0.043)
VGG-19-TL	0.821	0.823	0.902	0.859	0.897	0.911	0.916	0.912	0.864	0.872	0.896	0.872
	(0.015)	(0.026)	(0.000)	(0.011)	(0.005)	(0.019)	(0.034)	(0.005)	(0.012)	(0.029)	(0.049)	(0.011)
DANN-SL-TU	**0.687**	0.791	**0.683**	**0.726**	0.820	0.842	0.840	0.726	**0.741**	0.774	**0.728**	**0.744**
	(0.026)	(0.089)	(0.085)	(0.023)	(0.006)	(0.025)	(0.013)	(0.023)	(0.027)	(0.072)	(0.093)	(0.031)
Source	Nepal Earthquake											
Target	Matthew Hurricane				Ecuador Earthquake				Ruby Typhoon			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
VGG-19-SL	0.642	0.916	0.455	0.606	0.873	**0.908**	0.852	0.878	0.699	**0.890**	0.483	0.622
	(0.052)	(0.035)	(0.078)	(0.077)	(0.006)	(0.026)	(0.026)	(0.006)	(0.039)	(0.036)	(0.078)	(0.077)
VGG-19-TL	0.821	0.823	0.902	0.859	0.923	0.908	0.957	0.930	0.864	0.872	0.896	0.872
	(0.015)	(0.026)	(0.000)	(0.011)	(0.010)	(0.040)	(0.023)	(0.008)	(0.012)	(0.029)	(0.049)	(0.011)
DANN-SL-TU	**0.706**	0.860	**0.634**	**0.724**	0.871	0.860	**0.909**	0.884	**0.800**	0.808	**0.812**	0.819
	(0.023)	(0.086)	(0.088)	(0.038)	(0.023)	(0.025)	(0.038)	(0.022)	(0.042)	(0.043)	(0.065)	(0.042)
Source	Matthew Hurricane											
Target	Ecuador Earthquake				Nepal Earthquake				Ruby Typhoon			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
VGG-19-SL	0.747	0.744	0.813	0.777	0.760	0.788	0.808	0.797	0.683	0.650	0.851	0.736
	(0.009)	(0.013)	(0.006)	(0.005)	(0.009)	()0.008)	(0.036)	(0.014)	(0.006)	(0.011)	(0.023)	(0.002)
VGG-19-TL	0.923	0.908	0.957	0.930	0.897	0.911	0.916	0.912	0.864	0.872	0.896	0.872
	(0.010)	(0.040)	(0.023)	(0.008)	(0.005)	(0.019)	(0.034)	(0.005)	(0.012)	(0.029)	(0.049)	(0.011)
DANN-SL-TU	0.761	0.747	**0.870**	**0.802**	0.755	0.784	0.805	0.793	0.681	0.669	0.774	0.713
	(0.038)	(0.054)	(0.037)	(0.023)	(0.023)	(0.012)	(0.073)	(0.030)	(0.024)	(0.038)	(0.103)	(0.032)
Source	Ruby Typhoon											
Target	Matthew Hurricane				Nepal Earthquake				Ecuador Earthquake			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
VGG-19-SL	0.741	0.817	0.748	0.778	0.731	**0.849**	0.658	0.741	0.740	**0.865**	0.618	721
	(0.018)	(0.016)	(0.027)	(0.022)	(0.009)	(0.010)	(0.008)	(0.008)	(0.018)	(0.016)	(0.027)	(0.022)
VGG-19-TL	0.821	0.823	0.902	0.859	0.897	0.911	0.916	0.912	0.923	0.908	0.957	0.930
	(0.015)	(0.026)	(0.000)	(0.011)	(0.005)	(0.019)	(0.034)	(0.005)	(0.010)	(0.040)	(0.023)	(0.008)
DANN-SL-TU	0.736	0.818	0.740	0.774	**0.763**	0.813	**0.777**	**0.792**	**0.809**	0.814	**0.840**	**0.825**
	(0.017)	(0.057)	(0.051)	(0.009)	(0.021)	(0.023)	(0.083)	(0.033)	(0.032)	(0.024)	(0.080)	(0.037)

Figure 2. Examples of images that are correctly classified or miss-classified by different networks, when Ruby Typhoon is used as target disaster and Ecuador Earthquake is used as source disaster. (Top) Images that are classified correctly as *damage* by DANN-SL-TU and VGG-19-TL, and miss-classified as *no-damage* by VGG-19-SL. (Middle) Images that are miss-classified as *damage* by DANN-SL-TU and VGG-19-TL, while VGG-19-SL correctly classifies them as *no-damage*. (Bottom) The first two images on the left are miss-classified by both DANN-SL-TU and VGG-19-SL, while VGG-19-TL correctly classifies image (h) as *damage* and image (i) as *no-damage*. The last two images on the right have label *no-damage* and are miss-classified by all networks as *damage*.

Table 3. Proxy $\mathcal{A}$-distance between different domains. In the 'After Adaptation' case, $(A \rightarrow B)$ means that the model was trained with A as source and B as target, while $(B \rightarrow A)$ means that the model was trained with B as source and A as target.

Domain A	Domain B	Before Adaptation	After Adaptation $(A \rightarrow B)$	After Adaptation $(B \rightarrow A)$
Matthew Hurricane	Ruby Typhoon	1.287	0.625	0.141
Ecuador Earthquake	Matthew Hurricane	1.633	0.125	0.141
Ecuador Earthquake	Ruby Typhoon	1.637	0.250	0.219
Nepal Earthquake	Ecuador Earthquake	1.831	0.132	0.091
Nepal Earthquake	Matthew Hurricane	1.951	0.078	0.281
Nepal Earthquake	Ruby Typhoon	1.915	0.112	0.288



(a) Before adaptation (b) After adaptation (c) Before adaptation (d) After adaptation

Figure 3. Two dimensional representation of source (blue) and target (red) using the VGG-19-SL representation of the images (before adaptation), and the DANN-SL-TU transformed representation of the images (after adaptation), for two pairs of disasters. The pair on the left consists of two dissimilar disasters, Ecuador Earthquake (source, blue) and Ruby Typhoon (target, red). The pair on the right consists of two similar disasters, Ecuador Earthquake (source, blue) and Nepal Earthquake (target, red).The two representations are reduced to 2 dimensions using the t-SNE (Van Der Maaten 2014) technique.

with the quantitative evaluation in Table 2, these examples suggest that more unlabeled images from the target, and ideally more labeled images from the source, can potentially improve the results.

To evaluate the difficulty of the domain adaptation tasks in our study, we used the proxy $\mathcal{A}$-distance (Ben-David et al. 2007) to measure the domain shift. We expect domain adaptation to give better results than VGG-19-SL if the source and target domains are different, and similar results if the target and source domains are similar. Table 3 shows the domain pairs used in our experiments and their corresponding domain divergence in terms of the proxy $\mathcal{A}$-distance, before and after feature adaptation. To compute the proxy $\mathcal{A}$-distance, we trained a CNN model to separate the source domain X_S from the target domain X_T using unlabeled data. Then, we calculated the proxy $\mathcal{A}$-distance as $2(1 - 2\epsilon)$, where ϵ is the mis-classification error on test data. A high proxy $\mathcal{A}$-distance means that the two domains are far apart, and benefit from adaptation. A proxy $\mathcal{A}$-distance close to zero means that the two domains have essentially the same distribution. In addition to the proxy $\mathcal{A}$-distance before adaptation, we also calculated the proxy $\mathcal{A}$-distance after adaptation to understand if domains with different distributions are brought closer through adaptation. We should note that the model used to obtain the proxy $\mathcal{A}$-distance between $T(X_S)$ and $T(X_T)$ is equivalent with a two layer perceptron network, based on our model architecture. In practice, we found that sometimes the classification error may be larger than 0.5, which resulted in a negative value for the proxy $\mathcal{A}$-distance. In such cases, we took the absolute value of the proxy $\mathcal{A}$-distance, as suggested in (Ben-David et al. 2007). From Table 3, we can see that the distance between Matthew Hurricane and Typhoon Ruby is the smallest among all the distances, suggesting that the distributions of the two disasters are similar. Also, considering the pairs which contain Nepal Earthquake, we can see that the smallest distance is obtained for Ecuador Earthquake. Thus, the distances in Table 3 generally are in agreement with the classification results in Table 2. It is also interesting to note that, after domain adaptation, all proxy $\mathcal{A}$-distances become smaller, which means that the distributions of $T(X_S)$ and $T(X_T)$ are more similar as compared to the original distributions of the source and target datasets.

VISUAL DISTRIBUTION ANALYSIS

It can be proven that $T(X_S)$ and $T(X_T)$ have similar distributions after the domain adversarial training. Here, we visualize $T(X_S)$ and $T(X_T)$ for some sample source/target pairs to visually analyze the distributions of the transformed source and target data, by comparison with the distribution of the original source and target data.

To perform this visualization, first, we randomly selected 64 images from Ruby Typhoon and 64 images from Ecuador Earthquake, two disasters with relatively high distance. Using the VGG-19 model trained on Ecuador

Earthquake, we extracted the weights of the last fully connected layer in the VGG-19 model (this layer has dimension 4096). Next, we used t-SNE (Van Der Maaten 2014) to reduce the dimensionality from 4096 to 2. As a result, each image is represented by a two dimensional vector. We plotted the 128 images using the binary representation in Figure 3 (a), where we used blue triangles for images from Ecuador Earthquake and red dots for images from Ruby Typhoon. Similarly, using the DANN model trained on Ecuador Earthquake as source and Ruby Typhoon as target, we extracted the weights of the first fully connected layer, which represents the output of the transfer function $T(x)$. The graph for the 128 images (from Ecuador Earthquake and Ruby Typhoon) is shown in Figure 3 (b), where the blue triangles represent images from Ecuador Earthquake and the red dots represent images from Ruby Typhoon. As can be seen, the graph in Figure 3 (a) shows a clear boundary between the two domains, while in the graph in Figure 3 (b) the two domains are harder to separate as their distributions overlap significantly. Thus, the visualization of the reduced representations shows that after domain adaptation, the difference between source (Ecuador Earthquake) and target (Ruby Typhoon) has been indeed reduced.

In addition to performing the visualization of the transformed representations for two domains with high distance, we also performed if for two domains with relatively smaller distance. Specifically, we randomly selected 64 images from Nepal Earthquake and 64 images from Ecuador Earthquake. Following the same procedure described above, we plot two graphs in Figure 3 (c) and (d), respectively. In (c), the VGG-19 model was trained on Ecuador Earthquake and in (d), the DANN model was trained on Ecuador Earthquake as source and Nepal Earthquake as target. As can be seen in (c), the distributions of the two domains are indistinguishable even before adaptation, as the two domains have similar distributions in the first place. Thus, in this situation, DANN can not improve much the results of VGG-19, as the domains are already similar.

RELATED WORK

Domain adaptation on image data has received much attention in recent years (Gopalan et al. 2011). In particular, approaches based on Convolutional Neural Network (CNN) have been very successful, for example, Siamese networks (Bromley et al. 1993) such as the one proposed in (Long et al. 2015). Approaches based on reconstruction have also become popular. For example, Ghifary et al. (2016) applied image reconstruction on target unlabeled data by using a CNN autoencoder. Together with the reconstruction model, a supervised model was also trained on the source labeled data. The resulting model was used to retrieve information from target data. Ganin et al. (2016) applied adversarial training to adapt data from a source domain to a target domain. The adversarial training was originally used to generate synthetic images (Goodfellow et al. 2014). In domain adaptation, the adversarial training learns a data transformation which makes the source and target data to have similar distributions. Then, the classifier trained on the transformed source data can be used on the target data.

In disaster response, domain adaptation is a desirable approach, as it enables fast reaction when a disaster occurs. Several studies have applied domain adaptation approaches on disaster related text data. For example, H. Li et al. (2017) used the iterative Self-Training (Yarowsky 1995) strategy, with Naive Bayes as a base classifier, to perform domain adaptation for identifying tweets related to a disaster. Domain adversarial training has been applied to disaster tweet classification (Alam, Joty, et al. 2018). Compared to the research on disaster tweet classification, research on disaster image classification and analysis is more limited as of now. Bica et al. (2017) performed a visual analysis of two 2015 Nepal Earthquakes, and found a positive correlation between damage severity and the number of geotagged images posted by eyewitnesses of the earthquakes. Lagerstrom et al. (2016) studied the ability of machine learning approaches to identify Twitter images posted during a bush fire emergency situation, and concluded that the fire images can be identified with high accuracy.

More recently, Alam, Ofli, et al. (2018) and Mouzannar et al. (2018) recently published text and image multi-modal datasets, which have the potential to advance the state-of-the-art in disaster image analysis. Furthermore, Mouzannar et al. (2018) developed a deep learning approach to classify images based on damage. Similarly, Nguyen, Ofli, et al. (2017) used an approach based on CNNs to classify disaster images according to damage severity. Given that damage is a concept quantifiable on a continuous scale, X. Li et al. (2018) used the CAM approach to localize and quantify damage in disaster images. While these studies represent important steps towards disaster images analysis in real time, they assume target labeled data is available. Domain adaptation approaches, while greatly needed, have not been studied until now.

CONCLUSIONS

In this paper, we studied the application of domain adaptation to identify disaster images that show damage. We adapted the DANN approach by combining it with the VGG-19 model, and thus taking advantage of the available labeled data used to train VGG-19. Given that the proposed approach does not require labeled data from the target

domain, it can be used to identify damage images in real time, and this in turn can lead to faster disaster response. Experimental results suggest that the domain adaptation approach is especially useful when the source and target disasters are of different type, and thousands of labeled instances are available for the source disaster. As part of future work, we plan to apply the DANN approach to other disaster image classification problems, and also to study other domain adaptation approaches in the context of image analysis for disaster response.

REFERENCES

Aipe, A., Mukuntha, N., Ekbal, A., and Kurohashi, S. (2018). "Deep Learning Approach towards Multi-label Classification of Crisis Related Tweets". In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2018)*. Rochester, NY.

Alam, F., Joty, S., and Imran, M. (2018). "Domain Adaptation with Adversarial Training and Graph Embeddings". In: *arXiv preprint arXiv:1805.05151*.

Alam, F., Ofli, F., and Imran, M. (2018a). "CrisisMMD: Multimodal Twitter Datasets from Natural Disasters". In: *Proc. of the International AAAI Conference on Web and Social Media (ICWSM)*. Stanford, California, USA.

Alam, F., Ofli, F., and Imran, M. (2018b). "Processing Social Media Images by Combining Human and Machine Computing during Crises". In: *International Journal of Human-Computer Interaction* 34.4, pp. 311–327.

Bang, D. and Shim, H. (2018). "Improved Training of Generative Adversarial Networks Using Representative Features". In: *arXiv preprint arXiv:1801.09195*.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2007). "Analysis of representations for domain adaptation". In: *Advances in neural information processing systems*, pp. 137–144.

Bica, M., Palen, L., and Bopp, C. (2017). "Visual Representations of Disaster". In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. CSCW '17. Portland, Oregon, USA, pp. 1262–1276.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). "Signature Verification Using a "Siamese" Time Delay Neural Network". In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. NIPS'93. Denver, Colorado: Morgan Kaufmann Publishers Inc., pp. 737–744.

Caragea, C., Silvescu, A., and Tapia, A. H. (2016). "Identifying Informative Messages in Disasters using Convolutional Neural Networks". In: *13th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Rio de Janeiro, Brasil, May 22-25, 2016.*

Derczynski, L., Meesters, K., Bontcheva, K., and Maynard, D. (2018). "Helping Crisis Responders Find the Informative Needle in the Tweet Haystack". In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2018)*. Rochester, NY.

Enenkel, M., Saenz, S. M., Dookie, D. S., Braman, L., Obradovich, N., and Kryvasheyeu, Y. (2018). "Social Media Data Analysis and Feedback for Advanced Disaster Risk Management". In: *Social Web in Emergency and Disaster Management*.

Frej, W. (2018). "Hurricane Florence Flood Victims Turn To Social Media For Rescue". In: *HuffPost*.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). "Domain-adversarial training of neural networks". In: *JMLR* 17.1, pp. 2096–2030.

Ghifary, M., Kleijn, W. B., Zhang, M., Balduzzi, D., and Li, W. (2016). "Deep reconstruction-classification networks for unsupervised domain adaptation". In: *European Conference on Computer Vision*. Springer, pp. 597–613.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems*, pp. 2672–2680.

Gopalan, R., Li, R., and Chellappa, R. (2011). "Domain adaptation for object recognition: An unsupervised approach". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, pp. 999–1006.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Huang, Q. and Xiao, Y. (2015). "Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery". In: *ISPRS Int. Journal of Geo-Information* 4.3, pp. 1549–1568.

Imran, M., Chawla, S., and Castillo, C. (2016). "A Robust Framework for Classifying Evolving Document Streams in an Expert-Machine-Crowd Setting". In: *Proc. of the 18th Int. Conf. on Data Mining (ICDM)*. Barcelona, Spain.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

Kryvasheyeu, Y., Chen, H., Obradovich, N., Moro, E., Van Hentenryck, P., Fowler, J., and Cebrian, M. (2016). "Rapid assessment of disaster damage using social media activity". In: *Science advances* 2.3.

Lagerstrom, R., Arzhaeva, Y., Szul, P., Obst, O., Power, R., Robinson, B., and Bednarz, T. (2016). "Image Classification to Support Emergency Situation Awareness". In: *Frontiers in Robotics and AI* 3, p. 54.

Li, H., Caragea, D., Caragea, C., and Herndon, N. (2017). "Disaster Response Aided by Tweet Classification with a Domain Adaptation Approach". In: *Journal of Contingencies and Crisis Management (JCCM), Special Issue on HCI in Critical Systems*. 26.1, pp. 16–27.

Li, X., Zhang, H., Caragea, D., and Imran, M. (2018). "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.

Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015). "Learning Transferable Features with Deep Adaptation Networks". In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, pp. 97–105.

MacMillan, D. (2017). "In Irma, Emergency Responders' New Tools: Twitter and Facebook". In: *The Wall Street Journal*.

Mazloom, R., Li, H., Caragea, D., Imran, M., and Caragea, C. (2018). "Classification of Twitter Disaster Data Using a Hybrid Feature-Instance Adaptation Approach". In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2018)*. Rochester, NY.

Mouzannar, H., Rizk, Y., and Awad, M. (2018). "Damage Identification in Social Media Posts using Multimodal Deep Learning". In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2018)*. Rochester, NY.

Neppalli, V. K., Caragea, C., and Caragea, D. (2018). "Deep Neural Networks versus Naïve Bayes Classifiers for Identifying Informative Tweets during Disasters". In: *Proceedings of the 15th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2018)*. Rochester, NY.

Nguyen, D. T., Ofli, F., Imran, M., and Mitra, P. (2017). "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*. ASONAM '17. Sydney, Australia: ACM, pp. 569–576.

Nguyen, D. T., Alam, F., Ofli, F., and Imran, M. (2017). "Automatic image filtering on social networks using deep learning and perceptual hashing during crises". In: *arXiv preprint arXiv:1704.02602*.

Nguyen, D. T., Al-Mannai, K., Joty, S. R., Sajjad, H., Imran, M., and Mitra, P. (2017). "Robust Classification of Crisis-Related Data on Social Networks using Convolutional Neural Networks". In: *11th International AAAI Conference on Web and Social Media (ICWSM)*. Montreal, CA.

Pan, S. J. and Yang, Q. (2010). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Rhodan, M. (2017). "'Please Send Help.' Hurricane Harvey Victims Turn to Twitter and Facebook". In: *Time*.

Sen, A., Rudra, K., and Ghosh, S. (2015). "Extracting situational awareness from microblogs during disaster events". In: *Communication Systems and Networks (COMSNETS), 2015 7th International Conference on*. IEEE, pp. 1–6.

Simonyan, K. and Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Van Der Maaten, L. (2014). "Accelerating t-SNE using tree-based algorithms." In: *Journal of machine learning research* 15.1, pp. 3221–3245.

Vieweg, S., Hughes, A. L., Starbird, K., and Palen, L. (2010). "Microblogging during two natural hazards events: what twitter may contribute to situational awareness". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, pp. 1079–1088.

Villegas, C., Martinez, M., and Krause, M. (2018). "Lessons from Harvey: Crisis Informatics for Urban Resilience". In: *Rice University Kinder Institute for Urban Research*.

Wang, M. and Deng, W. (2018). "Deep Visual Domain Adaptation: A Survey". In: *Neurocomputing*.

Yarowsky, D. (1995). "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods". In: *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*. ACL '95. Cambridge, Massachusetts: Association for Computational Linguistics, pp. 189–196.