Nazr-CNN: Fine-Grained Classification of UAV Imagery for Damage Assessment

Nazia Attari*, Ferda Ofli, Mohammad Awad, Ji Lucas and Sanjay Chawla

Qatar Computing Research Institute
Hamad bin Khalifa University, Doha, Qatar
Email: *nazia.ec@gmail.com, {fofli, mhasan, jlucas, schawla}@hbku.edu.qa

Abstract—We propose Nazr-CNN[1], a deep learning pipeline for object detection and fine-grained classification in images acquired from Unmanned Aerial Vehicles (UAVs) for damage assessment and monitoring. Nazr-CNN consists of two components. The function of the first component is to localize objects (e.g. houses or infrastructure) in an image by carrying out a pixel-level classification. In the second component, a hidden layer of a Convolutional Neural Network (CNN) is used to encode Fisher Vectors (FV) of the segments generated from the first component in order to help discriminate between different levels of damage.

To showcase our approach we use data from UAVs that were deployed to assess the level of damage in the aftermath of a devastating cyclone that hit the island of Vanuatu in 2015. The collected images were labeled by a crowdsourcing effort and the labeling categories consisted of fine-grained levels of damage to built structures. Since our data set is relatively small, a pre-trained network for pixel-level classification and FV encoding was used. Nazr-CNN attains promising results both for object detection and damage assessment suggesting that the integrated pipeline is robust in the face of small data sets and labeling errors by annotators. While the focus of Nazr-CNN is on assessment of UAV images in a post-disaster scenario, our solution is general and can be applied in many diverse settings. We show one such case of transfer learning to assess the level of damage in aerial images collected after a typhoon in Philippines.
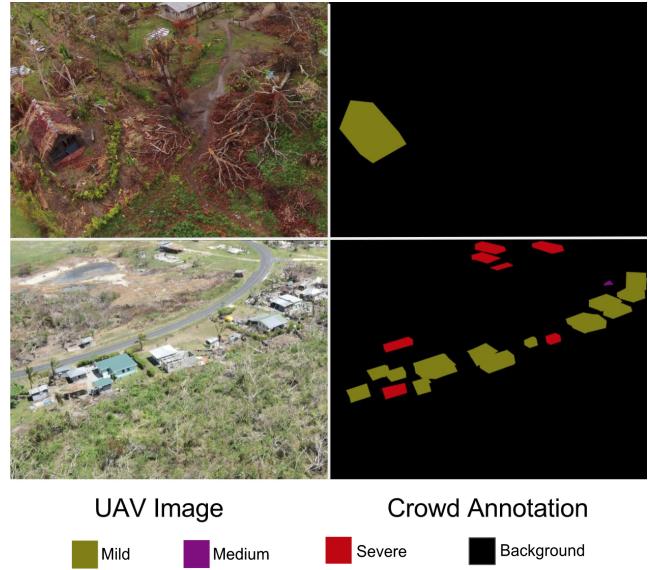
Fig. 1. Disaster images at different angle and elevation. Left: images-from-UAV. Right: Crowd annotations. Notice the difficulty of correctly annotating the images and thus generating accurate ground truth.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are now being increasingly used for structural damage assessment during routine monitoring and in the aftermath of natural disasters. For example, the use of UAVs for monitoring electrical power lines is gaining prominence, and so is their importance for damage inspection post a natural disaster event [1]. In fact, both the United States Federal Emergency Management Agency (FEMA) and the European Commission's Joint Research Center (JRC) have noted that aerial imagery will play an important role in disaster response and present a big data challenge [2].

As a specific example, the World Bank cooperated with the Humanitarian UAV Network (UAViators)[2] in the wake of Cyclone Pam, a category five cyclone that caused extensive damage in Vanuatu in March 2015, to carry out a post-disaster assessment of Vanuatu. The workflow followed was analogous to that of AIDR[3] (Artificial Intelligence for Disaster Response [3]), a system developed by QCRI to harness information from real-time tweets collected from an area struck by a natural disaster to help coordinate humanitarian relief activities. The acquired UAV images were annotated by a group of volunteers using MicroMappers[4], which is a crowdsourcing platform also built by QCRI in partnership with United Nations and the Standby Task Force, specifically for crisis management. Each annotator was asked to demarcate houses using a polygonal region and then associate a label with each region indicating the severity of damage (i.e., mild, medium and severe).

Analyzing large volumes of high-resolution aerial images generated after a major disaster remains a challenging task in contrast to the ease of acquiring them due to low operational costs. A popular approach is to use a hybrid strategy where initially a crowdsourcing effort is carried out to create a labeled training set which is used to infer a machine learning (ML) model [3]. The ML model then automatically classifies incoming images. However, until now, the image classification task (unlike text classification) suffered from low accuracy and lack of robustness in an uncontrolled environment.

---

[1]Nazr means "sight" in Arabic.
[2]http://uaviators.org/
[3]http://aidr.qcri.org/
[4]http://www.micromappers.org/

Before we describe our solution, we enumerate some of the challenges that must be overcome for object detection and fine-grained classification in images acquired from UAVs:

1) A single UAV image usually contains a high number of objects at different scales belonging to multiple categories. Also, the background in the image is often highly heterogeneous, e.g., ocean, sky, forest, grassland, etc. In Figure 1, we show a couple of example UAV images from our data set which confirm not only the difficulty of object detection task but also challenges in annotation. In the top image, the shack is almost indistinguishable from the land both in terms of texture and color.

2) The amount of labeled UAV imagery data is limited and the labels are often noisy since the damage assessment task is profoundly subjective. Furthermore, labeling small objects whose damage type is difficult to gauge makes the annotation task a lot more tedious. In our particular case, there is a significant conflict between built structures whose damage levels are rated as *medium* or *severe* (see Figure 6 for example images).

3) A key success of deep learning is that "feature engineering" is part of the learning process. However, we have observed, at least in our data set, distinguishing images based on texture is an important aspect of the problem and straightforward application of CNN is unlikely to result in fine-grained object classification.

A. Solution in a Nutshell

Our solution (**Nazr-CNN**) combines two deep learning pipelines. The first pipeline carries out a pixel-level classification (often known as semantic segmentation) to localize objects (segments) in images. The localized objects are then passed through a pre-trained Convolutional Neural Network (CNN) and a Fisher Vector (FV) encoding is extracted from the single (last) convolutional layer to generate a new representation of the objects. The FVs are then trained using a standard SVM classifier. This results in a highly accurate detection and fine-grained discrimination of houses based on their damage levels.

The rest of the paper is structured as follows. In Section II we precisely define and state the problem of fine-grained object classification. In Section III, we introduce the building blocks of **Nazr-CNN** with a particular focus on the use of FV encoding. In Section IV, we present the experiments and results. We survey the related work in Section V, and conclude the paper with a summary and future directions in Section VI.

II. PROBLEM STATEMENT

We now define the problem of object detection and classification in UAV images for damage assessment.

Given: A set of images obtained from UAVs over a disaster assessment area and labeled using a crowdsourcing platform. Built structures (e.g. houses) are labeled as *little-to-no damage/mild (M), partially damaged/medium (Md) and fully destroyed/severe (S)*.

Design: A machine learning classifier which takes as input an unlabeled image and classifies regions in the image as background (B) or containing structures which can be classified as M, Md or S.

Constraints: (i) Images often contain large number of objects with different damage types; (ii) the size of the data set is relatively small and (iii) there is disagreement among the annotators on the class labels.

III. DEEP LEARNING FRAMEWORK

Since our labeled data set is relatively small (1,085 images), we have built our deep learning pipeline (**Nazr-CNN**) using existing pre-trained networks as building blocks. Our proposed pipeline consists of integrating pixel-level classification (often known as semantic image segmentation) and texture discrimination. For semantic segmentation we have used DeepLab [4], a CNN network followed by a fully-connected Conditional Random Field (CRF) for smoothing. For texture discrimination we have used FV-CNN [5], which consists of extracting Fisher Vectors from a hidden layer of a pre-trained CNN. Intuitively, the aim of semantic segmentation is to localize objects (i.e., houses) in an image and the aim of texture discrimination is to distinguish between different types of damage and background. For a comprehensive background on CNNs, we refer the reader to the upcoming book by the pioneers of the field [6].

A. Pre-Trained Networks

In practice, CNNs are rarely fully trained afresh for new data sets because it is relatively rare to have access to large data sets–which is indeed the case we have. A popular choice of a pre-trained network is the VGG-16 network [7] that is trained on the ImageNet data set which contains over 1.2 million images and 1,000 categories (labels). The VGG-16 network consists of 16 layers and over 140 million weight parameters. There are two ways that a pre-trained network is used on new data sets. The first approach is to just pass each data point from the new data set and use the layers as feature extractors where each data point can be mapped into a new representation. Lower layers of VGG-16 can be considered as low-level feature extractors which should be applicable across domains. Higher layers tend to be more domain-specific. The second approach is to use the existing weights of the pre-trained network as an initialization for the new data set. This approach can often prevent overfitting but is computationally expensive.

B. Semantic Segmentation

We have used the DeepLab [4] to carry out pixel-level classification of images (i.e., semantic segmentation). This promotes the localization of objects in images. DeepLab combines the VGG-16 network with a fully-connected Conditional Random Field (CRF) model on the output of the final layer of CNN. The CRF overcomes the poor localization property of

CNNs and results in better segmentation. The CRF optimizes the following energy function:

$$E(\mathbf{x}) = \sum_i g_i(x_i) + \sum_{ij} h_{ij}(x_i, x_j) \tag{1}$$

where $\mathbf{x}$ is the pixel-level label assignment and $g_i = -\log P(x_i)$ is the label assignment probability at pixel i computed by CNN. The pairwise potential is given by

$$h_{ij}(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w_m \dot{k}^m(\mathbf{f_i}, \mathbf{f_j}) \tag{}$$

Here μ is the binary Potts model function, k^m is a Gaussian kernel for label k and is dependent on the features $\mathbf{f}$ extracted at the pixel level. The CRF is fully connected, i.e., there is one pair-wise term for each pair of pixels irrespective of whether they are neighbors or not. The reason that DeepLab uses fully connected CRF is that the objective is to extract local structure (shape) from the pixels and not just carry out a local smoothing (which might smooth out the local shapes). Pixel-level classification is an integral part of **Nazr-CNN**. Besides identifying the houses in UAV images and their shapes, pixel-level classification, as we will see in Section IV, serves as an automatic data cleaning step: it is robust against annotation errors of both kinds, i.e., it can identify segments which were missed by the annotator as well as fix labeling errors.

C. Fisher Vector Encoding

We use FV-CNN [5] to extract features which help in distinguishing between different levels of damage. In particular, we explain why the use of FVs as a representation of the input can help distinguish between different levels of damage. Since our data set is of modest size (1,085 labeled images), as is the usual practice, we use FV-CNN based on VGG-M architecture pre-trained on the ImageNet ILSVRC 2012 data set.

Fisher Vectors (FV) are a generalization of the popular Bag-of-Visual words (BoV) representation of images and are known to result in substantial increase in accuracy for image classification tasks [8]. As we will show that FVs extracted from high level CNN features are particularly useful for distinguishing different types of building damages.

We assume that an appropriate layer of CNN will generate a set of features $X = \{x_i, i = 1, \ldots, N\}$ where each $x_i \in \mathbb{R}^D$. Further we assume that the set X is generated from a Gaussian Mixture Model (GMM). Thus for each $x \in X$,

$$P(x|\lambda) = \sum_{i=1}^{K} w_i N(x, \mu_i, \Sigma_i) \tag{2}$$

and

$$\forall i \; : w_i \geq 0, \qquad \sum_{i=1}^{K} w_i = 1 \tag{3}$$

Here $N(x, \mu, \Sigma)$ is a multi-dimensional Normal (Gaussian) distribution. In order to avoid enforcing the constraints on the weights (w), a re-parameterization is carried out such that

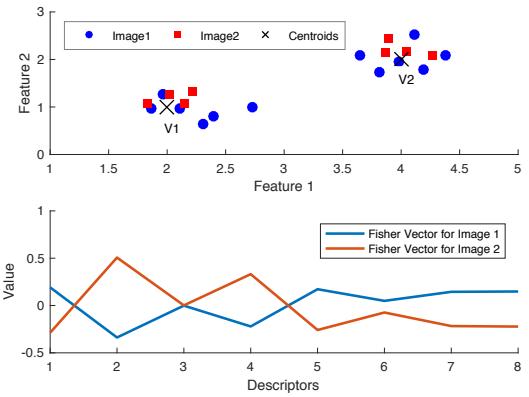$$w_i = \frac{exp(\alpha_i)}{\sum_{j=1}^{K} exp(\alpha_j)} \tag{4}$$

Fig. 2. Bag of Visual Words (BoV) vs. Fisher Vector (FV) representation.

Now the FV encoding of an $x \in X$ are the gradients of $\log P(x|\lambda)$ with respect to $\lambda = \{\alpha_j, \mu_j, \Sigma_j, j = 1, \ldots K\}$. The covariance matrix Σ_j is assumed to be a diagonal matrix. Thus

$$\nabla_{\alpha_j} \log P(x) = \gamma(j) - w_j \tag{5}$$

$$\nabla_{\mu_j} \log P(x) = \gamma(j) \left(\frac{x - \mu_j}{\sigma_j^2} \right) \tag{6}$$

$$\nabla_{\sigma_j} \log P(x) = \gamma(j) \left[\frac{(x - \mu_j)}{\sigma_j^3} - \frac{1}{\sigma_j} \right] \tag{7}$$

where the responsibility $\gamma(j)$ (the posterior probability) that x belongs to $N(u_j, \Sigma_j)$ is given by

$$\gamma(j) = \frac{w_j N(x, u_j, \Sigma_j)}{\sum_{i=1}^{K} w_i N(x, u_i, \Sigma_i)} \tag{8}$$

Often further normalization is carried out in Equations 5, 6, and 7 to arrive at the precise Fisher Vectors. Further details are provided in [8].

Example: We now give a simple example to show why FV encoding results in more discriminative features compared to the BoV model. Consider the example show in Figure 2. Assume we are given two-dimensional descriptors of two images (red and blue in the figure). The BoV model is to cluster the descriptors using the k-means algorithm and use the centroid as a representation of a visual word. In the example there are two visual words. Then each image is represented as a histogram consisting of counts associated with the visual words. For example the histogram of the blue image is $(6, 6)$ as six descriptors of the blue image are associated with the first visual word and six with the second visual word.

In contrast, the FV of the image has a dimensionality equal to the number of parameters of the GMM. For example the bottom plot in Figure 2 shows a parallel plot of eight of the ten features of each image. The FV is more sensitive to local variations (as the value at each descriptor represents the deviation from the GMM model). This is particularly suitable for texture discrimination where variation within a region and not the shape of the region is an important parameter.
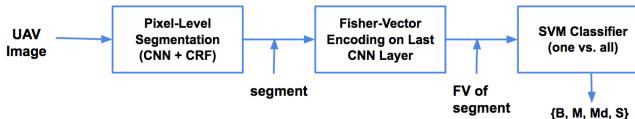
Fig. 3. **Nazr-CNN** combines pixel-level classification with FV-CNN. The Fisher Vectors are then trained using multi-class SVM.

D. Proposed Pipeline

We now present our fine-grained image classification pipeline (**Nazr-CNN**) for damage assessment in UAV images. Figure 3 shows the work flow of the proposed pipeline.

Training:

1) For pixel-level classification, the DeepLab system requires the image along with the masks created by the annotators.
2) The DeepLab system generates segments. Each segment is assigned with a label based on the ground truth mask with the maximum overlap.
3) The segment output from DeepLab is then fed into FV-CNN. For each segment, Fisher Vectors from an intermediate hidden layer of FV-CNN will be generated. Along with the Fisher Vectors, the label of the segment forms an element of the training set of the multi-class SVM. An SVM model is then induced from the segment and its label.

Testing:

1) An image (without annotation) is passed through DeepLab to generate segments.
2) Each segment is fed into FV-CNN, which generates Fisher Vectors for the segments.
3) The Fisher Vector of each segment is classified by the SVM model into one of the damage categories, i.e., B, M, Md, S.

IV. EXPERIMENTS AND RESULTS

In this section, we report on the extensive set of experiments that we have carried out to assess the performance of **Nazr-CNN**. We begin by describing in Section IV-A the data acquisition process used in our experiments. Then, we report the following four results: (i) the baseline accuracy of FV-CNN on pre-labeled segments to discriminate between severity of damage (Section IV-B), (ii) the baseline accuracy of DeepLab on ground truth annotations to localize and classify damaged houses (Section IV-C), and (iii) the accuracy of **Nazr-CNN** which is the combined pipeline of pixel-level semantic segmentation (with cross-validation) and Fisher Vector encoding using FV-CNN along with precision-recall computation (Section IV-D), and finally, (iv) the result of transfer learning on a novel dataset (Section IV-E).

A. Data Description

The UAV image data and corresponding damage annotations were acquired as part of an initiative by the World Bank in collaboration with the Humanitarian UAV Network (UAViators) during Cyclone Pam in Vanuatu in 2015. The
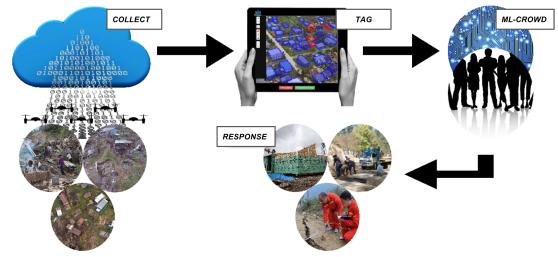

Fig. 4. UAV image acquisition and annotation workflow implemented in MicroMappers for this study, following the approach used for text classification by AIDR[6]. However, annotation of disaster images is substantially more difficult and until now, there did not exist an accurate and robust object detection and classification model.

workflow followed was analogous to that of AIDR (Artificial Intelligence for Disaster Response [3]) for text data, and is shown in Figure 4: images were acquired through UAVs and a group of digital volunteers using MicroMappers annotated built structures found in the images as described below. The resulting image dataset contains 3,096 images but approximately 65% of them do not contain any built structures or ground truth annotations. Hence, we use only a set of 1,085 images where each image contains one or more regions with different levels of damage: Mild (M), Medium (Md), Severe (S); and everything else is considered as Background (B). In addition to this data set, we have 60 images from a typhoon disaster that affected areas in Philippines to test the generalization potency of **Nazr-CNN**.

1) UAV Image Annotation: Images have been labeled using an online annotation platform and polygons are drawn around each built structure (sometimes there are more than one building in a single picture; all of them are traced). The different damage levels used in the annotation task include:

- *Severe (Fully Destroyed)*: A building should be considered fully destroyed if you see one or more of the following:
  - More than 50% (half) of the building is damaged
  - 2 or more walls are destroyed
  - Roof is missing or completely destroyed
- *Medium (Partially Damaged)*: A building should be considered partially damaged if you see one or more of the following:
  - About 30% of the building is damaged
  - Damage to 1 or 2 walls, or 1 wall fully destroyed
  - Roof still there but perhaps damaged
- *Mild (Little-to-no damage)*: A building should be considered little-to-no damage if 0% to 10% of the building is damaged.

Thereafter, majority voting is done (from multiple annotations) to obtain the final label. Note that different buildings may be tagged with different damage levels in the same image. This annotation process resulted in a total of 2,979 segments corresponding to damaged structures (i.e., houses) in 1,085

---

[6]Because of the cyclone the Internet was down and thus AIDR could actually not be activated to process tweets.
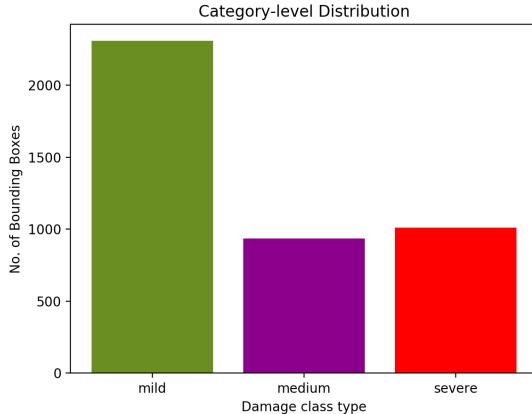
Fig. 5. The bar chart shows the damage-class distribution among 4253 bounding boxes in a total of 1085 image dataset.

GT vs. Pred	Mild	Medium	Severe
Mild	84±0.88	13±0.60	3±1.17
Medium	17±1.33	69±1.18	12±0.51
Severe	8±1.50	23±1.31	67±0.73

images. Figure 5 shows the class distribution, i.e. mild (54%), medium (22%) and severe (24%). This appears to be situation of high class imbalance, almost half of the dataset is covered by mild category and the rest is divided between the other two difficult categories (almost) equally. We perform 5-fold cross validation in all of the experiments and report performance in terms of mean and standard error.

2) Label Disambiguation: As mentioned earlier, the labels were annotated by crowd and there is a huge subjective incongruence which makes labels inconsistent thereby affecting learning. Also, the ground-truth is quite noisy since crowd-labeled polygons do not have crisp region boundaries. Figure 6 highlights few cases where difference in annotators point of view hold true. We can clearly see coarse annotations for the second image from the top; and rest others show lack in labeling consensus. Due to this reason, the overall performance of the system is hampered to some extent. We now report on the result of the first experiment.

B. Evaluation of FV-CNN as a Baseline

In this section we evaluate the performance of FV-CNN [5] for carrying out fine-grained damage assessment assuming that the ground-truth region is known and leveraging texture features using Fisher encoding. FV-CNN uses a pre-trained VGG-M model. The hidden layer used is a 512-dimensional local feature vector which gets further pooled into a Fisher Vector representation with a Gaussian Mixture Model (GMM) of size 64. The total resulting dimensionality is around 65K ($64 + 2 \times 64 \times 512$). We assume each component of the GMM has a diagonal covariance matrix. Finally, the region descriptors are classified using one-vs-all Support Vector Machines with a regularization hyperparameter $C = 1$.

Despite the severe imbalance of damage type segments, FV encoding based classification gave good results across classes. The confusion matrix in Table I show that regions with mild damage are classified with high accuracy. Medium and severe damage categories have lower accuracy. It is important to note that there was substantial disagreement between the annotators

about the level of damage suffered by built structures and the accuracy was exacerbated due to the class-imbalance problem. We use FV-CNN for further analysis mainly for two reasons: (i) our problem is closely related to texture analysis, and (ii) FV encoding is quite powerful in discriminating objects based on texture, and has achieved state-of-the-art performance in several texture benchmarks [5]. FV-CNN experiments were run using the *MatConvNet* toolbox on a Tesla K20Xm 6GB GPU card.

C. Pixel-level Segmentation by DeepLab as a Baseline

While the performance of FV-CNN for distinguishing between different damage levels based on texture is high, our aim is not only to recognize the type of damage but also to identify regions in the image where the damage occurred. Therefore, FV-CNN cannot be a solution by itself for our problem because it assumes the availability of the segments (bounding boxes) at test time. For this reason, we used DeepLab [4], a deep Convolutional Neural Network with a Conditional Random Field (CRF) layer on top, to segment the images so that the regions of potential damage can be recognized. The DeepLab system is one of the top-performing models on the PASCAL VOC-2012 semantic image segmentation task, reaching 71.6% mean average precision (mAP) and has thus become a natural choice for semantic classification. We used the "DeepLab-LargeFOV" model to perform per-pixel classification for the given images generating segments belonging to one of the damage categories (or background). CRFs were used as post-processing step to discern local shapes and overcome the noisy labeling of the annotators. Additionally, to overcome the significant class-imbalance, we modified the cross-entropy loss function using class-weighting [9].

Table II shows the cross-validation results for our data with and without class weighting. Though, the original paper suggests *mAP* is a better evaluation metric, however given the quality of annotations for our images (they are neither weak nor strong but somewhat of intermediate quality), we have used mean pixel-wise accuracy as a baseline for evaluating the results in **Nazr-CNN**. It is important to note that class-weighting enhances the localization and discriminative capability of the DeepLab system and improves the overall mean accuracy. DeepLab semantic segmentation experiments were run using the *Caffe* framework on an NVIDIA K3100M 4GB GPU memory card with a batch size of 4.

In Figure 7 we have provided a few example images which highlight that DeepLab is good in identifying damage regions but poor at texture discrimination. Additionally, the
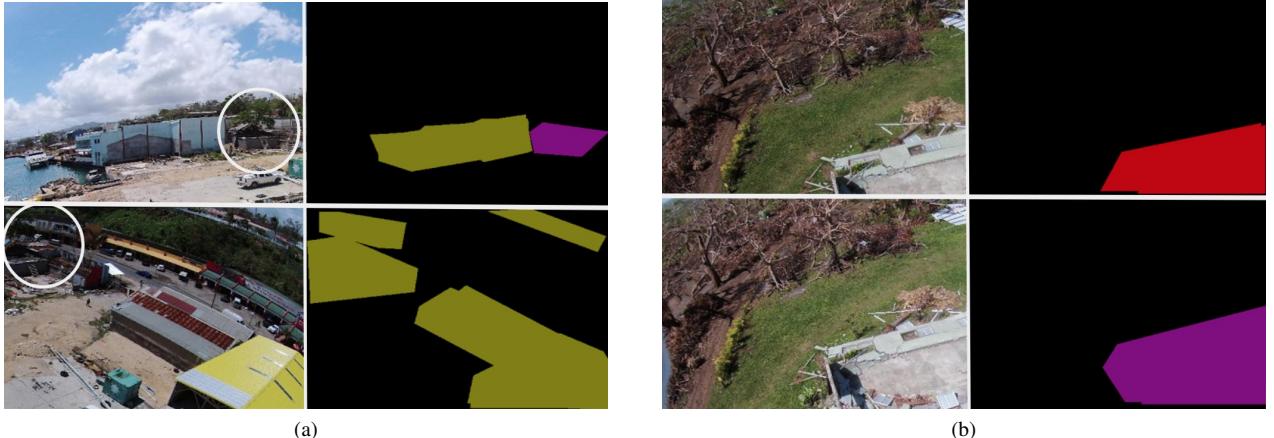
Fig. 6. Few examples of disagreement in annotation: (a) Here, two images contain same house (circled) but with contrasting labels. (b) Another case where annotators agree on damage but on varied severity scale.
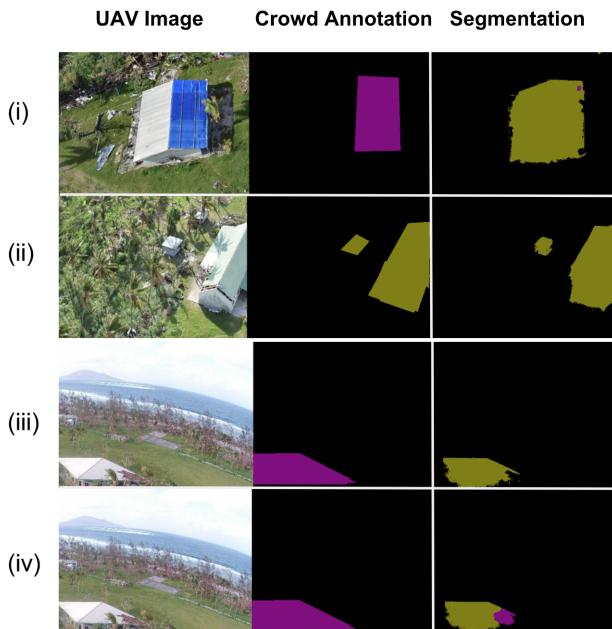

Fig. 7. Semantic segmentation results by DeepLab [4]: (i)-(ii) shows that DeepLab is good at identifying damage regions but weak at texture discrimination; (iii)-(iv) reflects class-weighting enhances the localization and discriminative capability of the DeepLab system.

poor performance is partly due to the fact that we have noisy ground-truth annotations.

D. Proposed Pipeline

Nazr-CNN combines pixel-level segmentation and fisher encoding. The accuracy of the combined system is shown in Table III. The average accuracy of the DeepLab (X) system is shown in the first column. It is clear that class weighting improves the overall accuracy. In the second column the standalone accuracy of FV-CNN (Y) is shown. Thus if we combine the two pipelines, the hypothetical best accuracy is a product of X and Y which is shown in column three. However

Nazr-CNN does slightly better then a simple product of the two. This suggests that in **Nazr-CNN** the FV-CNN component is able to fix some of the errors incurred by DeepLab. The overall performance of **Nazr-CNN** is limited by segments obtained by DeepLab; but DeepLab is affected due to poor annotations and multiple objects.

Figure 8 shows the detection and classification results on a few of the images with results from semantic segmentation and **Nazr-CNN**. The examples shown in the figure highlight some of the following cases:

(a) both models perform equally well,
(b) labeling is poor but models are robust,
(c) DeepLab is capable of identifying damage regions while Fisher Vector encoding helps in texture recognition.
(d) the hard cases in identifying more than approximately 10-15 objects in a single image.

The example images demonstrate that **Nazr-CNN** tends to combine the shape and texture features in a successful manner, and provides a reasonably robust results. In addition to reporting the average accuracy, we also compute average precision-recall values for semantic segmentation (DeepLab) and **Nazr-CNN** (DeepLab + FV-CNN) in Tables IV and V, respectively. We also compute the values with and without class weighting. The numbers show that precision drops with class weighting but recall improves for semantic segmentation. On the other hand, for **Nazr-CNN** precision values almost remains unchanged but again recall improves. In both tables, the achieved performance is lowest for the *medium* damage category.

E. Transfer Learning

To test the effectiveness of **Nazr-CNN** and the need to create the high-performance learner for a target domain from a similar source domain, we performed transfer learning. We evaluated our model on the 60 images of disaster-struck (typhoon) areas in Philippines. The terrain in Philippines is quite different from that of our original data set; it is more like
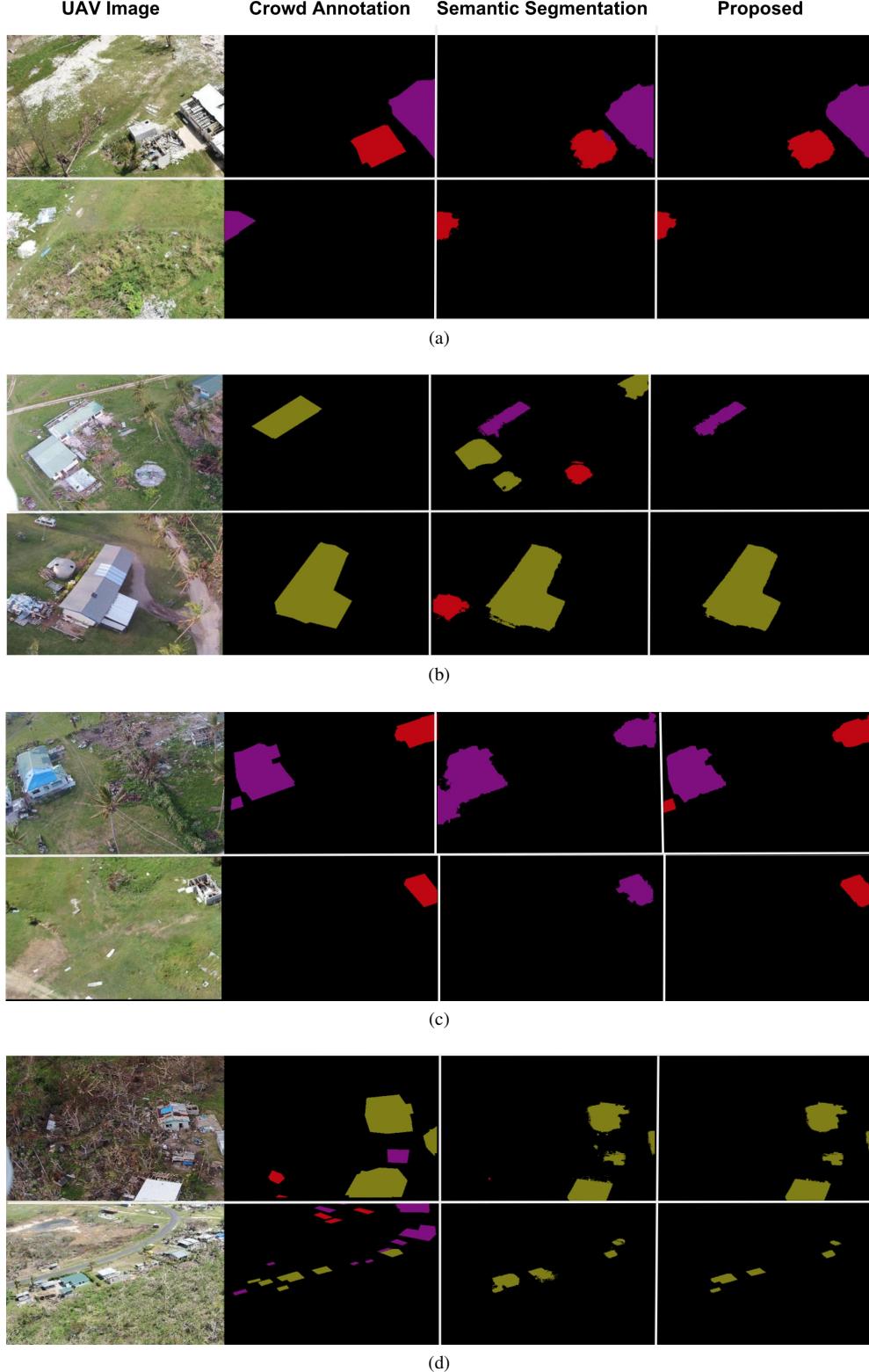
Fig. 8. Examples to evaluate semantic segmentation [4] and our proposed pipeline (**Nazr-CNN**): (a) Two models are equally good. Also, the bottom image highlights that both algorithms are powerful in capturing the obvious severe damage in the UAV image which was originally labeled as medium, thereby reflecting label inconsistency. (b) Both examples show extra segments identified by semantic segmentation which are not annotated by crowd. This affects the evaluation due to improper ground truth in these cases. (c) **Nazr-CNN** performs well in differentiating between severe and medium. Learns texture in an efficient manner. (d) Hard cases in identifying more than approximately 10-15 objects in a single image.

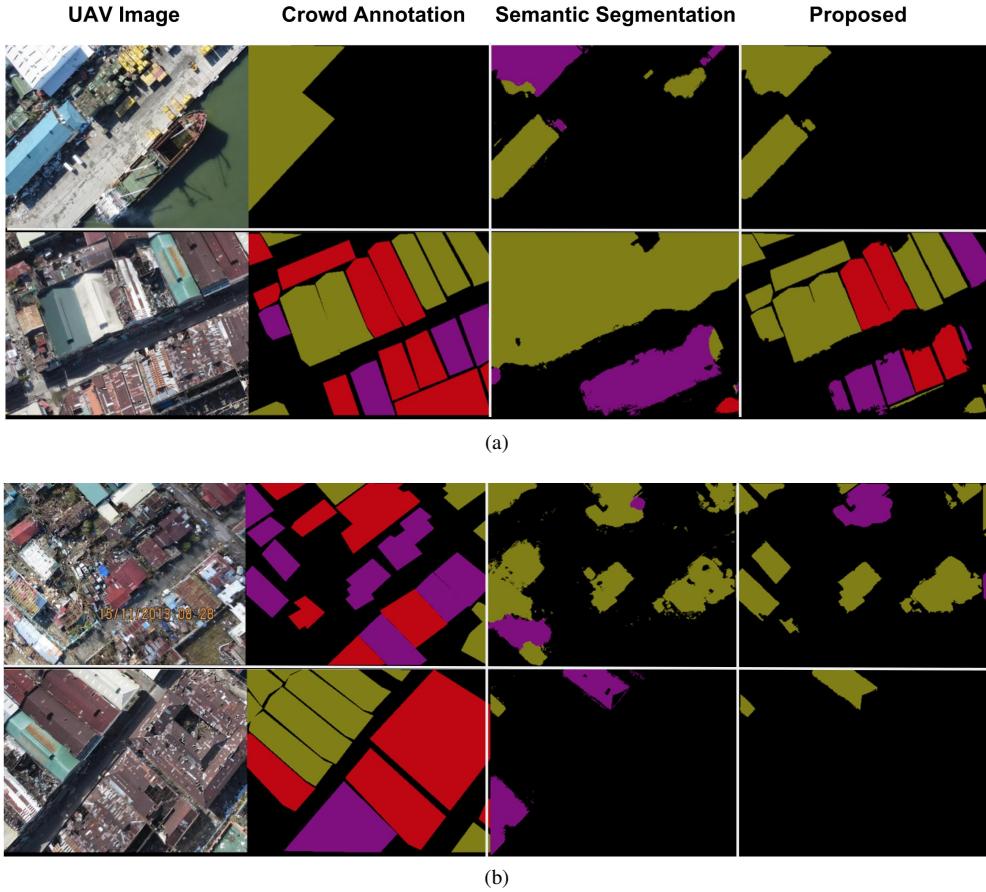| UAV Image | Crowd Annotation | Semantic Segmentation | Proposed |

(a)

(b)

Fig. 9. Transfer learning on Philippines data: (a) Both images show that our proposed pipeline (**Nazr-CNN**) can discriminate texture to some degree. (b) Exhibit the difficult case (top image) as well as poor prediction (bottom). The type of damage in this case is quite different and more difficult to categorize; often the roofs are also of same color and material type.

a city landscape. The results in Table VI show that **Nazr-CNN** performs relatively better and improves on mild and severe classes than the baseline semantic segmentation (DeepLab). This is further exhibited in Figure 9 which shows the texture discriminatory potential to some degree. It also draws attention to the cases where the models perform poorly due to the fact that the damage is observed mostly on the roofs with same color and differentiating medium from severe proves to be very challenging. Furthermore, the low number for severe category is mainly due to imbalanced data as most of the buildings come under mild category.

V. RELATED WORK

Deep Learning techniques now underpin many computer vision tasks. In particular the best algorithms for image classification are now likely to be based on Convolutional Neural Networks (CNNs) [10]–[16]. CNNs (especially with the pooling layer) are designed to be translation invariant. However one of the key strengths of CNNs, that they are designed to be invariant to spatial transformations, is precisely the weakness when it comes to object localization as required for damage assessment from UAVs. To overcome the weakness of CNNs, Chen et.al. [4], have introduced the "DeepLab" system which combines CNNs with Conditional Random Fields (CRFs). CRFs are particularly useful for capturing local interaction between neighboring pixels. In particular DeepLab performs pixel-level classification, a task sometimes known as semantic image segmentation (SS) in the computer vision community. Early attempts in semantic image segmentation used a set of bounding boxes and masked regions as input to the CNN architecture to incorporate shape information into the classification process to perform object localization and semantic segmentation [14], [17]–[20]. Taking a slightly different approach, some studies employed segmentation algorithms independently on top of deep CNNs that were trained for dense image labeling [21], [22]. More direct approaches, on the other hand, aim to predict a class label for each pixel by applying deep CNNs to the whole image in a fully convolutional fashion [23], [24]. Similarly, [9] and [25] train an end-to-end deep encoder-decoder architecture for multiclass pixelwise segmentation.

One of the key strengths of systems based on deep learning is that they automatically infer a representation of the data suitable for the defined task. For example, the lower layers of deep learning correspond to a representation suitable for low-level vision tasks while the higher layers are more domain specific [6] and obviates the need for pre-defined

TABLE II
SEMANTIC SEGMENTATION RESULTS BY DEEPLAB [4]

Damage Level	Mean Accuracy(%)
3-class	59.04±1.12
3-class*	63.07±1.71

* indicates training with class weighting

TABLE III
COMPARISON OF ALL THE MODELS. **Nazr-CNN** PERFORMS WELL IN DIFFERENTIATING TEXTURE, AND IMPROVES ON CLASSIFYING SEGMENTS IDENTIFIED BY DEEPLAB

Damage Levels	Segmentation (X)	FV-CNN (Y)	Hypothesis (X*Y)	**Nazr-CNN**
3-classes	59.04±1.12	83.6±0.86	49.35±0.96	50.25±1.51
3-classes*	63.07±1.71	83.6±0.86	52.71±1.47	53.41±2.04

* indicates training with class weighting

TABLE IV
PRECISION-RECALL FOR DEEPLAB PER DAMAGE CATEGORY

Damage Levels	Precision		Recall	
	w/o cls wt	w/ cls wt	w/o cls wt	w/ cls wt
Mild	78.83±2.97	66.87±1.49	62.42±3.33	73.90±4.20
Medium	59.86±2.63	50.67±3.19	43.04±5.06	43.70±5.72
Severe	78.88±6.21	67.76±7.05	39.83±8.21	48.85±4.76

TABLE V
PRECISION-RECALL FOR **Nazr-CNN** PER DAMAGE CATEGORY

Damage Levels	Precision		Recall	
	w/o cls wt	w/ cls wt	w/o cls wt	w/ cls wt
Mild	86.02±5.06	85.59±5.50	64.34±2.64	74.66±3.42
Medium	56.38±6.98	56.96±7.29	30.44±11.46	36.08±10.21
Severe	79.24±11.21	79.64±6.89	38.85±8.68	44.73±6.96

TABLE VI
EVALUATION OF TRANSFER LEARNING ON PHILIPPINES DATA

Model Architecture	Overall Accuracy (%)	Per-class Accuracy (%)		
		Mild	Medium	Severe
Segmentation	36.26	62.80	35.70	10.30
Nazr-CNN	**40.90**	86.77	17.01	16.50

feature engineering like SIFT [26]. Fisher Vectors (FV) are an important representation used in computer vision for object discrimination [8]. FVs generalize the Bag of Visual Word (BoV) model which are now often built on top of CNN hidden layers. In particular FV-CNN [5], is a recent attempt to combine the use of CNN and FVs for texture recognition and segmentation.

Aerial image analysis for detecting objects, classifying regions and analyzing human behavior is an active research area. A recent overview is presented in Mather and Koch [27] which also mentions the use of damage assessment datasets (e.g., [28]) as a benchmark. Works which use texture for aerial imagery include [29]. Examples of other recent work include Blanchart et al. [30], where they utilize SVM-based active learning to analyze aerial images in a coarse-to-fine setting. Bruzzone and Prieto [31] is an example of a change detection-based analysis technique. Zhang et al. [32] develop coding schemes for classifying aerial images by land use. Similarly, Hung et al. [29] tackle with weed classification based on deep auto-encoders while Quanlong et al. [33] analyze the urban vegetation mapping using random forests with texture-based features. Oreifej et al. [34] recognize people from aerial images. Gleason et al. [35] and Moranduzzo and Melgani [36] detect cars in aerial images using kernel methods and support vector machines.

There are also studies that aim to produce a complete semantic segmentation of the aerial image into object classes such as building, road, tree, water [37]–[39]. Some of the recent attempts apply deep CNNs to perform binary classification of the aerial image for a single object class [37], [40], [41]. These recent attempts to apply deep learning techniques to high-resolution aerial imagery have resulted in highly accurate object detectors and image classifiers, suggesting that automated aerial imagery analysis systems may be within reach.

However, most of the aforementioned aerial image analysis methods assume that the images are captured at a nadir angle via satellites with known ground resolution, and hence, fixed viewpoint and scale for the objects in the scene. However, UAVs usually fly at variable altitudes and angles, and therefore, capture oblique images with varying object sizes and appearances. Therefore, in contrast to the traditional aerial image analysis and computer vision paradigms, a new set of computer vision and machine learning approaches must be developed for UAV imagery to account for such differences in the acquired image characteristics.

Finally, there are a few other recent damage assessment studies on images collected from social media platforms, not necessarily UAV images though. For example, [42], [43] tackle with fire detection whereas [44] addresses flood detection from social media images. Furthermore, [45] proposes an automatic image processing pipeline for social media imagery data, and [46], [47] further explore infrastructural damage assessment problem, mainly at the image classification level.

VI. SUMMARY AND FUTURE WORK

In this paper we have proposed an integrated deep learning pipeline (**Nazr-CNN**) for identifying built structures (e.g., houses) in UAV images followed by a fine-grained damage classification. The images were collected in the aftermath of a natural disaster with the aim of assessing the level of damage.

Nazr-CNN has two distinct components. The first component carries out a pixel-level classification of images (a task often known as semantic segmentation) with the aim of identifying damaged structures in an image. The aim of the second component is to carry out a fine-grained classification of the structures identified to assess the severity of damage. We use a Fisher Vector representation of image segments to assess the severity of damage. **Nazr-CNN** is particularly robust against noisy labels and appears to be height invariant–a necessary property for UAV images.

To the best of our knowledge this is the first known deep learning pipeline for object detection and classification of UAV images collected from disaster struck regions. At this point,

our work handles a more complex problem, such as, image segmentation for noisy aerial images. We plan to further investigate end-to-end deep learning techniques to better handle the noise and ambiguity in ground truth annotations.

REFERENCES

[1] J. F. Galarreta, N. Kerle, , and M. Gerke, "UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning," *Natural Hazards and Earth Systems Science*, vol. 15, pp. 1087–1101, 2015.

[2] F. Ofli, P. Meier, M. Imran, C. Castillo, D. Tuia, N. Rey, J. Briant, P. Millet, F. Reinhard, M. Parkan, and S. Joost, "Combining human computing and machine learning to make sense of big (aerial) data for disaster response," *Big Data*, vol. 4, no. 1, pp. 47–59, Mar 2016.

[3] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "AIDR: Artificial intelligence for disaster response," in *International Conference on World Wide Web*, ser. WWW'14 Companion, 2014, pp. 159–162.

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR*, vol. abs/1412.7062, 2014.

[5] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3828–3836.

[6] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

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

[8] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[10] Y. LeCun and Y. Bengio, "The handbook of brain theory and neural networks," M. A. Arbib, Ed., 1998, ch. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., 2012, pp. 1097–1105.

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations (ICLR)*, April 2014.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015, pp. 1–9.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, June 2016.

[17] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*, Sep 2014, pp. 297–312.

[18] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 3376–3385.

[19] R. B. Girshick, "Fast R-CNN," in *International Conference on Computer Vision*, 2015, pp. 1440–1448.

[20] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[21] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[22] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[23] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2016.

[24] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *International Conference on Computer Vision*, Dec 2015, pp. 2650–2658.

[25] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[26] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[27] P. M. Mather and M. Koch, *Computer Processing of Remotely-Sensed Images: An Introduction*, no. v. 4.

[28] "Remote sensing damage assessment:," Tech. Rep., January 2010.

[29] C. Hung, Z. Xu, and S. Sukkarieh, "Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav," *Remote Sensing*, vol. 6, no. 12, 2014.

[30] P. Blanchart, M. Ferecatu, and M. Datcu, "Cascaded active learning for object retrieval using multiscale coarse to fine analysis," in *IEEE Int Conf on Image Processing*, Sept 2011, pp. 2793–2796.

[31] L. Bruzzone and D. F. Prieto, "An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images," *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 452–466, Apr 2002.

[32] H. Zhang, J. Zhang, and F. Xu, "Land use and land cover classification base on image saliency map cooperated coding," in *IEEE International Conference on Image Processing*, Sept 2015, pp. 2616–2620.

[33] Q. Feng, J. Liu, and J. Gong, "UAV remote sensing for urban vegetation mapping using random forest and texture analysis," *Remote Sensing*, vol. 7, no. 1, p. 1074, 2015.

[34] O. Oreifej, R. Mehran, and M. Shah, "Human identity recognition in aerial images," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 709–716.

[35] J. Gleason, A. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," in *IEEE International Conference on Robotics and Automation*, May 2011, pp. 2065–2070.

[36] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 3, pp. 1635–1647, March 2014.

[37] P. Dollar, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1964–1971.

[38] S. Kluckner and H. Bischof, "Semantic classification by covariance descriptors within a randomized forest," in *IEEE International Conference on Computer Vision Workshops*, Sept 2009, pp. 665–672.

[39] S. Kluckner, T. Mauthner, P. M. Roth, and H. Bischof, *Asian Conference on Computer Vision (ACCV)*, 2010, ch. Semantic Classification in Aerial Imagery by Integrating Appearance and Height Information.

[40] V. Mnih and G. E. Hinton, *European Conference on Computer Vision*, 2010, ch. Learning to Detect Roads in High-Resolution Aerial Images.

[41] V. Mnih and G. Hinton, "Learning to label aerial images from noisy data," in *Proceedings of the 29th Annual International Conference on Machine Learning (ICML 2012)*, June 2012.

[42] R. Peters and P. d. A. Joao, "Investigating images as indicators for relevant social media messages in disaster management," in *Int Conference on Information Systems for Crisis Response and Management*, 2015.

[43] S. Daly and J. Thom, "Mining and classifying image posts on social media to analyse fires," in *International Conference on Information Systems for Crisis Response and Management*, 2016, pp. 1–14.

[44] R. Lagerstrom, Y. Arzhaeva, P. Szul, O. Obst, R. Power, B. Robinson, and T. Bednarz, "Image classification to support emergency situation awareness," *Frontiers in Robotics and AI*, vol. 3, p. 54, 2016.

[45] F. Alam, M. Imran, and F. Ofli, "Damage assessment from social media imagery data during disasters," in *International Conference on Advances in Social Networks Analysis and Mining*, 2017, pp. 1–4.

[46] D. T. Nguyen, F. Alam, F. Ofli, and M. Imran, "Automatic image filtering on social networks using deep learning and perceptual hashing during crises," in *International Conference on Information Systems for Crisis Response and Management*, May 2017.

[47] D. T. Nguyen, F. Ofli, M. Imran, and P. Mitra, "Damage assessment from social media imagery data during disasters," in *International Conference on Advances in Social Networks Analysis and Mining*, 2017, pp. 1–8.