Convolutional neural networks for object detection in aerial imagery for disaster response and recovery

Yalong Pi, Nipun D. Nath, Amir H. Behzadan

Abstract

Accurate and timely access to data describing disaster impact and extent of damage is key to successful disaster management (a process that includes prevention, mitigation, preparedness, response, and recovery). Airborne data acquisition using helicopter and unmanned aerial vehicle (UAV) helps obtain a bird’s-eye view of disaster-affected areas. However, a major challenge to this approach is robustly processing a large amount of data to identify and map objects of interest on the ground in real-time. The current process is resource-intensive (must be carried out manually) and requires offline computing (through post-processing of aerial videos). This research introduces and evaluates a series of convolutional neural network (CNN) models for ground object detection from aerial views of disaster’s aftermath. These models are capable of recognizing critical ground assets including building roofs (both damaged and undamaged), vehicles, vegetation, debris, and flooded areas. The CNN models are trained on an in-house aerial video dataset (named Volan2018) that is created using web mining techniques. Volan2018 contains eight annotated aerial videos (65,580 frames) collected by drone or helicopter from eight different locations in various hurricanes that struck the United States in 2017–2018. Eight CNN models based on You-Only-Look-Once (YOLO) algorithm are trained by transfer learning, i.e., pre-trained on the COCO/VOC dataset and re-trained on Volan2018 dataset, and achieve 80.69% mAP for high altitude (helicopter footage) and 74.48% for low altitude (drone footage), respectively. This paper also presents a thorough investigation of the effect of camera altitude, data balance, and pre-trained weights on model performance, and finds that models trained and tested on videos taken from similar altitude outperform those trained and tested on videos taken from different altitudes. Moreover, the CNN model pre-trained on the VOC dataset and re-trained on balanced drone video yields the best result in significantly shorter training time.

1. Introduction

According to the United Nations Office for Disaster Risk Reduction (UNISDR), in the 10-year period ending in 2014, natural disasters have affected 1.7 billion people, claimed 700,000 lives, and cost 1.4 trillion Dollars in damages [1]. With the changing climate, the frequency and severity of natural disasters are also on the rise [2], requiring people and governments to be better prepared and equipped to cope with the effects of such catastrophes. Timely retrieval and integration of disaster information is critical for effective disaster management. Particularly, real-time spatial information about disaster damage and risk is of paramount importance to designing appropriate mitigation strategies and response plans [3]. Besides, identifying and locating objects of interest is a central issue in urban search and rescue (USAR) missions [4]. Recent advancements in personal handheld devices, mobile connectivity, and cloud storage, have created new opportunities for data collection and spatial mapping before, during, and after a disaster strikes. Among various methods of data collection, unmanned aerial vehicles (UAVs) or drones have drawn much attention due to their growing ubiquity, easy operation, large coverage area, and storage capacity. Compared to traditional methods of aerial data collection such as helicopter flyovers, UAV-based data collection is more affordable, less resource-intensive, and can provide high-resolution imagery from difficult-to-reach places. More importantly, with proper implementation, it can be outsourced to volunteer groups and ordinary citizens, thus reducing the workload on first responders and search and rescue (SAR) units, while enabling data collection at scale.

https://doi.org/10.1016/j.aei.2019.101009

Received 25 May 2019; Received in revised form 20 September 2019; Accepted 29 October 2019

1474-0346/ © 2019 Elsevier Ltd. All rights reserved.
However, some major challenges still exist that hinder the widespread use of drones in large-scale operations that involve people, communities, and urban assets. On the operational side, existing FAA regulations require the presence of a human operator and maintaining a clear line-of-sight (LOS) at all times during the mission. Moreover, FAA does not allow drone flights at night or over people, although some recent progress has been made to ease such restrictions [6]. On the logistical side, the current practice of drone data collection and processing is heavily human-centered and based on post-processing of captured data. A UAV reconnaissance team consists of at minimum three people each trained and certified to conduct a particular task; one person flies the drone, another person checks drone camera view, and the third person records and reports the status to first responders and SAR units.

The aim of this paper is to use artificial intelligence (AI) to address some of the problems associated with the logistics of carrying out drone missions in information-rich environments such as post-disaster operations. In particular, we introduce a fully annotated visual dataset called Volan2018, as well as train and test a convolutional neural network (CNN) architecture to detect, classify, and localize information on valuable ground assets in aerial drone views of natural disasters with a special focus on hurricanes. CNN models apply convolutional computations on input matrices (digital images and videos) to extract useful features (e.g., object edges, shapes, and patterns), and ultimately predict object classes and locations. The parameters of a CNN model are produced by training the model on a large annotated dataset, which involves selecting the weights of the neural network such that the error between ground truth and prediction is gradually minimized down to an optimum point. Using the trained model to detect, classify, and localize objects in drone views with high speed (real-time) and accuracy will help improve disaster information retrieval and exchange. Findings of this work are also sought to be a key component of information exchange platforms and decision support system (DSS) applications that integrate and share different data modalities with stakeholders involved in search, rescue, and recovery processes, ranging from ordinary people to first responders, law enforcement, local jurisdictions, insurance companies, and non-governmental organizations (NGOs).

2. Literature review

The digital revolution is characterized by a fusion of technologies that is blurring the lines between the physical, digital, and biological worlds, and has pushed the frontiers of science and discovery. In the last decade alone, the seamless integration of data into everyday life has resulted in exponentially large, multimodal datasets (e.g., photos, videos, blogs, and tweets) in the public domain such as online content sharing platforms and social networking sites. Within the context of natural disasters, when communities participate in data collection and information exchange, new opportunities can emerge for better understanding of urban vulnerabilities, capacities, and risks, as well as creating data-driven methods for damage assessment and recovery planning.

Several researchers have explored crowdsourced data collection. Craglia et al. [7] used user-generated content (UGC) such as Facebook and Twitter to produce volunteered geographic information (VGI) maps. Kim and Hastak [8] utilized machine learning (ML) and online social networks to obtain disaster-related information. Yuan et al. [9] proposed using semantic analysis of tweets to extract disaster information, and verified this method in hurricane Matthew. Goodchild and Glennon [10] investigated crowdsourcing geographic disaster information collection to the public in order to facilitate the information flow. More recently, researchers have been looking into the challenge of detecting ground objects from aerial drone footage. Baker et al. [11] proposed a Monte-Carlo algorithm for using UAVs to conduct post-disaster search, which was proven faster than the conventional sweeping search. Radovic et al. [12] used transfer learning based on the you-only-look-once (YOLO) algorithm to detect airplanes on the ground from aerial views. Han et al. [13] conducted real-time object detection in drone imagery using region-based CNN (R-CNN) and kernelized correlation filter (KCF). Narayanan et al. [14] demonstrated the possibility of using a high-performance cloud computer for real-time aerial detection from UAVs. Guirado et al. [15] employed Faster R-CNN to track and count the number of whales from satellite images. Previous work, however, has not systematically investigated and documented the problem of CNN-based aerial inspection in natural disasters.

In computer vision, the task of identifying objects in an image broadly includes four different steps of classification, detection, localization, and semantic segmentation. Image classification (i.e., predict the class to which an object belongs) has been developing rapidly since Russakovsky et al. [16] introduced ImageNet and Alex et al. proposed AlexNet [17]. Simonyan and Zisserman [18] introduced VGGNet that has 11–19 layers and achieves better accuracy in image classification than AlexNet by increasing the number of network layers. Szegedy et al. [19] proposed GoogLeNet, which includes inception modules that apply convolutional and max pooling at the same time, thus outperforming VGGNet. ResNet [20] used residual network in its image classification architecture that can outperform an average human by 3.57%. However, while these methods can determine the existence of an object class in an image, they fail to recognize the location(s) and number of detected objects (a.k.a., detection and localization). One intuitive method of localizing objects in an image is by sliding windows of different sizes across the image, and classifying the content of each window, a process that is extremely slow. Faster object detection and localization techniques use a parallel network that proposes smaller candidate regions in the image to locate multiple objects, followed by using classifiers to predict the class of those candidate regions. Past research in this domain includes Girshick et al. [21] who introduced R-CNN, which uses region proposal to replace the extremely slow sliding box method. Girshick et al. [22] later introduced Fast R-CNN using region of interest (RoI), which led to a considerable improvement of the prediction speed. Ren et al. [23] proposed Faster R-CNN, which adopts region proposal network (RPN) to reduce calculation time, and achieves 73.2% and 70.4% mean average precision (mAP) on PASCAL VOC [24] 2007 and 2012, respectively. Dai et al. [25] introduced FRCN that generates a 3 × 3 position-sensitive map of the input, and produces results close to Faster-RCNN but 2.5–20 times faster. Several one-stage detectors have been introduced recently with faster speeds than the above methods. For example, Redmon et al. [26] introduced YOLO, which reaches 45 frames per second (FPS) with a 63.4% mAP on VOC 2007. Liu et al. [27] proposed SSD based on MobileNet achieving 76.8% mAP on VOC 2007. Lin et al. [28] introduced RetinaNet which yields 37.8% COCO average precision (COCO-AP) on COCO [29] using focal loss to better learn hard examples.

It must be noted that there is often a trade-off between accuracy and speed, as demonstrated in Table 1. Real-time spatial information is important as it allows decision-makers to model scenarios and interact with the spatiotemporal dimensions of a disaster as it evolves [30]. Since the objective of this work is to enable time-sensitive disaster information retrieval in aerial views, the computational speed of the CNN model is considered a major constraint. The desired model must be able

<table>
<thead>
<tr>
<th>CNN architecture</th>
<th>COCO-AP (%)</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv2 [31]</td>
<td>21.6</td>
<td>25</td>
</tr>
<tr>
<td>SSD321 [27]</td>
<td>28.0</td>
<td>61</td>
</tr>
<tr>
<td>R-FCN [25]</td>
<td>29.9</td>
<td>85</td>
</tr>
<tr>
<td>DSSD513 [27]</td>
<td>33.2</td>
<td>156</td>
</tr>
<tr>
<td>FRCN [25]</td>
<td>36.2</td>
<td>172</td>
</tr>
<tr>
<td>RetinaNet-101-800 [28]</td>
<td>37.8</td>
<td>198</td>
</tr>
</tbody>
</table>

Table 1: COCO-average precision (COCO-AP) versus time (millisecond) on COCO test-dev dataset. adapted from [28]
to operate in real-time (i.e., 30 FPS or higher) and be light enough to run on ordinary drones. After careful examination of several CNN architectures, the authors have selected YOLO v2 [31] as a benchmark model. YOLO v2 is one of the fastest object detection algorithms that can achieve 78.6% mAP on PASCAL VOC 2007 [24] and 48.1% COCO-AP on COCO [29] test-dev datasets. Details of this network architecture will be presented in Section 5, following a discussion of the designed methodology (Section 3), and a description of the developed dataset (Section 4).

3. Methodology

Given the growing intensity and frequency of hurricanes in the continental U.S., and the nature of the destruction left behind (high visibility from the air), we use hurricane damage data as a motivating case with the long-term goal of adopting the designed approach to other disaster domains such as typhoons, earthquakes, wildfires, and landslides. By convention, the term “hurricane” describes a tropical cyclone that forms in the north Atlantic, northeastern Pacific, the Caribbean Sea, and the Gulf of Mexico [32]. The 2017 hurricane season was the costliest in the U.S. history with seventeen named storms including hurricanes Harvey, Irma, and Maria [33]. Hurricane forecasters predict fifteen named storms and eight hurricanes in 2019 [34]. When a hurricane strikes, the immediate damage could include fallen branches, uprooted trees, toppled power lines, roof failure, and wall collapse. The resulting storm surge may also lead to extensive flooding, submerged vehicles, flooded roads, trapped people and livestock, and damage to critical infrastructure such as power grid, chemical plants, and levees. The widespread extent of hurricane damage and the vast number of affected people and neighborhoods in coastal communities has been well documented over the years through videos taken by ordinary people, news channels, government agencies, and volunteer groups, and posted on web-based content sharing (e.g., YouTube) and social media (e.g., Facebook, Twitter) sites. Such visual content can be webmined and used for training and testing disaster-domain CNN models. Fig. 1 illustrates the overall methodology of this research, which is explained at length in the following Subsections.

3.1. Identification of ground objects of interest (GOIs)

A comprehensive literature review was first conducted to identify the most important categories of objects on the ground that are of interest to first responders and SAR units, and visible from the air. Pinpointing the location of such objects and their potential progression with time is the focal point of disaster management operations. For instance, in rescue planning, SAR units need to know the location and number of trapped people and submerged vehicles. In order for first responders and SAR units to navigate to the point of response, they need to have precise information about flooded roads and areas to avoid. Once floodwaters have receded, identifying debris locations is of the essence to municipalities to expedite cleanup and rebuild. Similarly, insurance companies and disaster aid agencies such as the U.S. Federal Emergency Management Agency (FEMA) may be interested in obtaining an overall damage assessment of a subdivision or neighborhood (by counting the number of damaged roofs or toppled trees) to start processing claims and allocating funds. Considering these and other actions that are routinely undertaken by different stakeholders to quantify disaster damage, a list of key ground objects of interest (GOIs) is compiled and presented in Table 2.

3.2. Dataset preparation

Web mining was implemented to extract video content from YouTube. To yield best results, the following keywords were used: hurricane, damage, drone, UAV, disaster, aftermath, and aerial. Videos that contained desirable classes (i.e., GOIs), as listed in Table 2, were then used to create an in-house annotated hurricane imagery dataset, named Volan2018. This dataset is further split into two subsets based on the viewpoint altitude of the video capturing platform. In particular, videos that were taken by a drone (flying at a relatively lower altitude of below 300 feet) were grouped into Volan2018-D (“D” for drone) dataset, and those taken by a helicopter (flying at a relatively higher altitude of above 1000 feet) were grouped into Volan2018-H (“H” for helicopter) dataset. This distinction later allowed us to test the effect of altitude on the accuracy of GOI detection. In creating the Volan2018 dataset (regardless of D or H designation), we also included videos from different geographical locations and hurricanes, as well as videos of different lengths and content richness (number of GOIs). This approach was necessary to verify whether the CNN model trained on a particular disaster or location can still produce satisfactory results when tested on a different disaster or location, thus supporting domain adaptation and generalizability. As explained in Section 4, each subset (D and H) was thoroughly annotated and post-processed (to ensure content balancing), and then split into training (60%), validation (20%), and testing (20%) datasets.

3.3. Model training, deployment, and performance

The YOLO v2 model was pre-trained on two large publicly available datasets, COCO [29] and VOC [24]. We used transfer learning to retrain the YOLO v2 [31] network (pre-trained on COCO and VOC) on the Volan2018 dataset. Past work has shown that through transfer learning, a CNN model can provide better and more consistent result on a relatively small dataset (a.k.a. target dataset) since it learns useful and relevant intermediate features from a large dataset (a.k.a. source dataset) [35,36]. For training, as explained in Section 5, several combinations are experimented based on video altitude (D or H), pre-trained dataset (COCO or VOC), and whether the training data is balanced (denoted by “B”) or unbalanced (denoted by “U”). These experiments are necessary to determine key factors affecting the performance of the CNN models. In total, eight training set combinations are created, namely D-B-COCO, D-B-VOC, D-U-COCO, D-U-VOC, H-B-COCO, H-B-VOC, H-U-COCO, and H-U-VOC. A detailed description of these combinations is provided in Section 5. Resulting CNN models are then tested on testing portions of Volan2018 dataset to output GOI positions (bounding boxes) and class labels. In addition, to examine the performance of the CNN models in GOI detection in aerial footage obtained from new locations, two additional unseen videos (captured from geographical regions not represented in Volan2018 dataset) are tested. For performance measurement, precision and recall analysis is carried out to quantify the discrepancy between ground-truth (annotated classes and bounding boxes) and model predictions (detected classes and bounding boxes). In addition, mAP, defined as the mean value of average precisions across all classes, is calculated for each model. Section 5 contains results of model performance and analysis.

---

**Fig. 1.** Research methodology.
4. Data description

4.1. Data collection

A content-rich, properly annotated visual dataset is key to accurate model training for disaster GOI detection. There are few aerial image recognition datasets, such as AID [37], a large classification dataset retrieved from Google Earth, which contains 10,000 images with 30 classes, Pattern Net [38] with 38 classes each containing 800 images for satellite-level object sensing, and Minidrone, an annotated drone video dataset created for surveillance in a parking lot with two main classes of humans and cars [39]. However, none of these datasets is semantically rich enough for natural disaster information collection, necessitating the need for creating a new dataset in this research, named Volan2018. As previously explained, videos included in Volan2018 were obtained from YouTube using a semi-supervised web mining technique, and collectively contain the GOIs listed in Table 2. Overall, Volan2018 contains eight aerial videos (with 1280 × 720 resolution, 30 FPS) from different hurricanes that occurred during the 2017–18 hurricane seasons (including hurricanes Harvey, Maria, Irma, and Michael). Together, this video dataset covers locations in the city of Houston (Texas), southern areas of Texas (i.e., Port Aransas, Holiday Beach, and Rockport), Puerto Rico (U.S. territory), Big Pine Key, Mexico Beach, and St Joe Beach (all in Florida). Table 3 presents detailed information on the Volan2018 video dataset.

As part of pre-processing, frames containing watermarks in Volan001 and 002, and black margins in Volan008 were removed. As mentioned in Section 3.2, considering the viewpoint altitude, Volan001, 002 and 003 are grouped as drone dataset (D), while Volan004, 005, and 006 are grouped as helicopter dataset (H). Also, Volan007 (drone video) and Volan008 (helicopter video) from hurricane Michael, which are different from Volan001 through Volan006, serve as completely unseen test videos to validate the real-world applicability of this work beyond common testing and validation practices in computer vision. While in traditional computer vision, the same (already available) dataset is split into training and testing portions, here the goal is to train a CNN model in advance using past disaster data, and then test it on new disaster data captured from a different time and location (i.e., new domain). In our experiments, we first train and test our models on Volan001 through Volan006 that cover six different locations in three hurricanes during the 2017 season (hurricanes Harvey, Maria, and Irma). We will then test the trained models (without retraining) on Volan007 and Volan008 captured from the 2018 season (hurricane Michael) at two completely different locations (St Joe Beach and Mexico Beach, Florida) to assess the scalability and generalizability of the trained models to new situations.

4.2. Dataset annotation

We annotate each frame of Volan2018 with DarkLabel [40] by drawing bounding boxes that individually cover each GOI listed in Table 2. One object in one frame is defined as one instance so that the pixel coordination values (with the most upper left point of the frame serving as origin) of the bounding boxes are obtained for training. Annotating consecutive frames in a video using DarkLabel can cost one annotator approximately 2s per frame, although this varies based on the interest that must be annotated. For example, to annotate a 10-minute long video (18,000 frames), 10 h of work is expected. Table 3 illustrates annotation samples for each class following the annotation strategy below:

- Flooded area: Flooded areas in Volan2018 are widespread and connected. Bounding boxes corresponding to flood can thus cover the most portion of a frame (including other class instances). To avoid this, we split the flooded area by drawing multiple small bounding boxes that cover only the flooded area without including other GOIs.
- Debris: We ignore small-sized debris in annotation since small amounts of debris will most likely not turn into major obstacles during disaster operations. Large-sized debris that could impede transportation or cause damage is annotated by drawing bounding boxes around the edges. Similarly, visible destructions to infrastructure and splintered houses are annotated as debris.
- Cars: Bounding boxes are drawn around the edges for each car, individually.
Vegetation: We draw large boxes covering the tree cluster if those boxes do not cover any other class instance. Otherwise, one separate box is drawn for each tree. Lawn is not classified as vegetation for the purpose of this annotation.

- Damaged roof: We draw bounding boxes that cover the entire roof (include undamaged part), but do not include elements below the roof (e.g., walls).

- Undamaged roof: We draw bounding boxes that reach the roof drip edge (containing skylight, chimney, etc.), but do not include elements below the roof (e.g., walls).

The instance distribution of Volan001 through 006 is shown in Fig. 3. In this Figure, each colored line represents one class, the x-axis indicates the frame number, and the y-axis indicates the number of instances. This distribution shows the variation of visible objects on the ground in camera’s viewpoint as the vehicle (drone or helicopter) flies over different areas. For example, in Volan005, a helicopter flies over flooded areas in frame 1 through 360, followed by areas that are not flooded in frames 361 through 1170, and then over flooded areas again in frames 1171 through 2460. A similar visual analysis can be performed for all classes in all six videos.

In order to quantify the class/instance diversity in Volan2018 dataset, we introduce two indices that are calculated for each of the videos contained in the dataset. These include Instance Number (IN) which refers to the total number of instances of each class (as listed in Table 3), and Instance per Frame (IPF) which represents the average number of instances of a particular class per frame, and calculated by Eq. (1). In this Equation, Frame Number (FN) refers to the number of frames that contain at least one instance of a particular class.

\[
\text{Instance Per Frame (IPF)} = \frac{\text{Instance Number (IN)}}{\text{Frame Number (FN)}}
\]

Table 4 lists the IPF values for each class in each of the eight videos in Volan2018. Comparing the IPF values for the same class across different videos (taken from different locations or at different times) reveals the significance of a certain type of damage or hazard across time (temporal scale) and locations (spatial scale), as viewed in aerial imagery taken by drone or helicopter. For example, Volan005 contains more flooded areas than other videos, while the location represented in Volan003 has suffered less damage (as indicated by the fewer number of damaged roofs) than other locations. The last column in Table 4 lists the average IPF value for the entire Volan2018 dataset. Evidently, the most frequent classes in Volan2018 are flooded area (5.73), undamaged roof (4.68), and vegetation (3.15) whereas debris, car, and damaged roof do not appear frequently. The average IPF can assist future data collection by directing more attention to classes that are under-represented (lower IPF values), thus helping balance the dataset.

We also calculate a third index, called damaged roof percentage (DRP) for each video, using Eq. (2) in which \(n_{UR} \) and \(n_{DR} \) are the total number of instances belonging to damaged roof and undamaged roof classes in each video, respectively. The DRP value reflects the extent of roof damage in a particular location, which could be a good indicator of the overall building damage in a neighborhood or subdivision, and help inform insurance-related decision-making (i.e., quick estimation of claims). As listed in Table 4, Volan004 has the highest DRP value, whereas Volan003 has nearly no damaged roofs. Overall, damage quantification using IPF and DRP indices can be of significant value to disaster management, particularly for response and recovery operations [41].

\[
\text{Damaged Roof Percentage (DRP)} \% = \frac{n_{UR}}{n_{UR} + n_{DR}}
\]

4.3. Data balancing

The CNN model was initially trained on the entire Volan2018 dataset. However, given the long duration and large number of instances in Volan003, this model was heavily biased toward Volan003, i.e., the model produced very accurate prediction on the test frames of Volan003 but performed poorly on test frames from other videos. To remedy the problem of unbalanced data in model training, there are three general strategies that include pre-processing, cost-sensitive learning, and a combination of both [42]. Considering crowdsourcing as a possible future direction of this work to scale up data collection by utilizing consumer-grade drones and low computational power, we favor pre-processing methods for data balancing which include over-sampling the instances of minority classes, and under-sampling the instances of majority classes [43]. In this research, to avoid overfitting we apply under-sampling to Volan2018 dataset. Prior to under-sampling, the balance ratio (BR) of each video in Volan2018 is calculated using Eq. (3), where \(N_k \) is the number of instances of class \(k \) in frame \(N \), and \(c \) and \(f \) are the total number of classes, and the total number of frames in each video, respectively. In addition, in this Equation, \(N = \sum_{k=1}^{c} \sum_{f=1}^{f} N_{k,f} \) represents the average number of instances per class. A higher BR value indicates a less balanced video. For instance, as shown in Table 5, Volan003 has a BR value of 0.79, making it the least balanced video among all eight videos in Volan2018 dataset. For reference, we also measure the BR for two popular datasets, namely COCO [29] with a BR of 0.84 and VOC [24] with a BR of 0.54, making VOC a more balanced dataset in comparison.

\[
\text{Balance Ratio (BR)} = \frac{\sum_{k=1}^{c} \sum_{f=1}^{f} N_{k,f} - \bar{N}}{c \star N}
\]

To perform under-sampling and balance the dataset with respect to the number of instances in each class, two values are defined and used: diversity balancing threshold (DBT) (Eq. (4)) and quantity balancing...
threshold (QBT) (Eq. (5)). Here, $c_k$ is the number of different classes present in frame $k$, $i_{n,k}$ is the number of instances of class $n$ in frame $k$, and $c$ and $f$ are the total number of classes and the total number of frames in each video, respectively.

\[
\min_{1 \leq k \leq f} (c_k) \leq \text{Diversity Balancing Threshold (DBT)} \leq \max_{1 \leq k \leq f} (c_k) \leq \frac{\min_{1 \leq n \leq c} \left( \min_{1 \leq k \leq f} i_{n,k} \right)}{\max_{1 \leq n \leq c} \left( \max_{1 \leq k \leq f} i_{n,k} \right)} \leq \text{Quantity Balancing Threshold (QBT)}
\]  

(4)

For any given video in Volan2018 dataset, we calculate DBT and QBT considering all of the frames ($1 \cdots f$) in that video. If frame

Fig. 3. Instance numbers for each class in Volan2018 Drone dataset (D) containing Volan001, 002, and 003, and Helicopter dataset (H) containing Volan004, 005, and 006.
contains distinct classes \( c_k \) fewer than the DBT value (i.e., \( \text{DBT}_k < \text{DBT} \)), the frame is excluded from further consideration (training and testing), in favor of under-sampling. In Volan2018, the default value of DBT is set to 1 to eliminate frames that contain no instances of any class. Next, we exclude any frame if the number of instances of a given classes in that frame is greater than QBT (i.e., \( n_k^f > \text{QBT} \)). This comparison is done for \( \text{QBT} = \{1, \text{max} \} \), and the BR corresponding to each QBT is computed. The QBT that yields the minimum BR, a.k.a., \( \text{BR}_{\text{min}} \) (hence, the most balanced dataset) is ultimately selected, and frames that pass the test are preserved. A summary of the calculations is shown in Table 5. Finally, for the two subsets (D or H), the same number of frames from each video (selected randomly with uniform probability), equal to the minimum number of remaining frames in all videos is selected for training. As shown in Table 5, this process leads to 925 frames per video in the D subset, and 519 frames per video in the H subset. In addition, from this Table, it is observed that after dataset balancing, Volan003 has the highest BR drop (from 0.79 to 0.24) and the largest frame loss (from 39,990 down to 925). At the conclusion of the balancing process, the original Volan2018 dataset is split into six parts, namely Drone Balanced (D-B), Drone Unbalanced (D-U), Helicopter Balanced (H-B), Helicopter Unbalanced (H-U), Volan007, and Volan008. These parts are used in different combinations to train, validate, and test the CNN models. It must be noted that Volan007 and Volan008 are considered unseen videos (not used for initial training and testing, and exclusively kept for scalability assessment) and therefore, are not balanced.

5. Experiments and results

In this Section, we first introduce the CNN model (YOLO v2) structure and transfer-learning scheme used for pre-training. Next, eight different models (based on different training and testing data combinations) and their performance are presented. Finally, a discussion of key factors that influence model performance is provided.

5.1. Model architecture and transfer learning

YOLO v2 [31] is a CNN model that takes red, green and blue (RGB) imagery as input, and outputs predictions in form of target objects’ classes and their coordinates in the image. Fig. 4 shows the architecture of YOLO v2, which has 23 layers including convolutional (CONV) and max-pooling (MAX) layers, each with kernel, normalization, and activation functions. A CONV layer has kernels with parameters in each kernel cell, and extracts features such as shape and color from the input. Each kernel applies convolutional computations by a stride to cover the entire image and outputs one channel. Taking the first CONV in Fig. 4 as an example, after applying 32 kernels with stride 1 on the input \((416 \times 416 \times 3)\), the output is a 32-channel matrix. Each layer computes the input and then passes the result on to the next layer until the last output layer. Altogether, the model divides the input image into a
13\times13 grid, and at the output layer each grid predicts 5 anchor boxes. Each anchor box contains x and y coordinates, width, height, confidence value (confidence with which the anchor box contains an object), and probability for each class. Since our model predicts six classes, in each anchor box, the output corresponding to a single image (i.e., video frame) contains 13\times13\times5\times(5+6)=9295 predicted values.

The key for a model to produce the correct prediction is the weights in the kernels. Training CNN models involves backpropagation, a process during which kernel weights are updated. In each iteration, the difference between ground truth and prediction is calculated by loss function, L, as laid out in Eq. (6). In this Equation, xi, yi, wi, hi, ci, and pi represent predicted x and y coordinates of the center of the box, width, height, objectness score, and class probability, respectively, while xi, yi, wi, hi, ci, and pi represent ground truth x and y coordinate of the center of the box, width, height, objectness score, and class probability, respectively, and box represents the box scale factor. The function BC(\hat{a}_i, a_i) (Eq. (7)) represents the binary cross-entropy where \hat{a}_i and a_i are ground truth and predicted values, respectively. During training, backpropagation updates the weights systematically using loss function and regression function to yield better prediction until the optimum point.
Initially, the YOLO v2 [31] is pre-trained on the ImageNet dataset [16]. Next, object detection weights are re-trained on either VOC [24] or COCO [29], and finally, transfer learning is used to train the model on Volan2018 dataset, as shown in Fig. 5. In particular, pre-trained model weights from ImageNet and COCO/VOC are first loaded in our model, and weights are further updated by re-training the model on Volan2018. For each subset (i.e., D-B, D-U, H-B, H-U), we split the dataset into training (60%), validation (20%), and testing (20%). The first 22 layers are initially frozen, and only the last layer is trained for 25 epochs with a learning rate of $10^{-3}$. Next, the entire model is fine-tuned with an initial learning rate of $10^{-4}$, which is gradually decreased by half if loss does not drop after 3 epochs. To avoid overfitting, training is terminated if the validation loss does not drop in 10 consecutive epochs.

### 5.2. Results

We created eight models (numbered 1 through 8) based on different combinations of pre-training, training and validation datasets, as listed in Table 6. The training process follows the steps described earlier. All models are ultimately tested on D-B, D-U, H-B, H-U test subsets, as well as on the entire length of Volan007 and Volan008 videos.

The performance metrics include intersection over union (IoU), precision, recall, mAP, and F1 score. IoU is defined by Eq. (8) and is equal to the overlapping area between detection and ground truth divided by their union area. Fig. 6 illustrates the concept of IoU. The best prediction is when IoU is 100%, i.e., prediction boxes and ground truth boxes exactly overlap. However, while 100% IoU is nearly impossible to achieve (given limitations of existing CNN architectures), an IoU value of 50% to 90% is commonly used in many computer vision applications. In this work, given the scope of work and complexity of disaster scenery, a detection is deemed successful (a.k.a., true positive or TP) if IoU $\geq 50%$.

$$\text{Intersection over Union (IoU)} = \frac{\text{Intersection Area}}{\text{Union Area}}$$

In order to calculate precision and recall, in addition to TP cases, predictions that result in false positive (FP), and false negative (FN) must also be considered. Examples of all three possible cases are shown in Fig. 7. Of note, in object detection, there is no finite number of true negative (TN) cases (i.e., there is no ground truth object in a particular region of the image and the model does not detect anything in that region) and, therefore, TN is not considered in the performance assessment. Precision and recall are calculated using Eqs. (9) and (10), and the harmonic average of precision and recall (a.k.a., F1 score) is obtained from Eq. (11).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
In the prediction stage, threshold (i.e., minimum class probability) is commonly used to eliminate low-confidence detection boxes and produce human understandable output. For example, a model with the threshold value of 0.1 only outputs detections with class probability higher than 0.1. To investigate the suitable threshold for real-world applications, we test Model 2 (trained on D-B-VOC) on Volan007 video, and record precision, recall, and F1 score for threshold values ranging from 0.0 to 0.9 with 0.1 increments. As later summarized in Table 7, Model 2 outperforms other models when tested on completely unseen drone footage. Sample results for classes debris and undamaged roof are shown in Fig. 8 and Fig. 9. According to this Figure, larger threshold values lead to lower recall and higher precision, indicating that the model produces fewer detections, therefore, fewer false positives and more false negatives. Theoretically, the best threshold is at the point where precision, recall, and F1 score converge. However, in cases where GOIs are highly valuable or not detecting them may have severe consequences (e.g., trapped people on the roof of a flooded home), high recall is necessary even with low precision. On the other hand, different information users may have different opinions about the value of different GOIs, thus leading to varying perceptions of precision and recall values. Besides, different classes and test data have different corresponding precision, recall and F1 curves. In conclusion, the selection of precision and recall, essentially the threshold value, primarily depends on the needs and expectations of the end user.

For testing the performance of the CNN models, cumulative precision and recall values for each class (a.k.a., GOI) are calculated and plotted in precision-recall curves. The area under the precision-recall curve for each class is then calculated as average precision (AP) for that class. Finally, the average of all AP values is reported as the mAP of the model. Results for all 8 CNN models are shown in Table 7. As indicated by these results, models trained on drone videos tend to perform better when tested on drone videos, while models trained on helicopter videos tend to perform better when tested on helicopter videos. This observation supports the influence of viewpoint altitude on model performance. The best mAP performance for D set and H set are 74.48% (Model 3 trained on D-U-COCO and tested on D-U) and 80.69% (Model 7 trained on H-U-COCO and tested on H-B), respectively. These promising results support that automatically extracting information from aerial views is feasible.

Although training and testing images in D and H are different, they are still extracted from similar videos that closely resemble one another (e.g., same disaster, or same location). In practice, the test video could be captured from an utterly different altitude, camera, flying speed, disaster event, location, time, and lighting condition. A robust model must perform adequately in detecting GOIs when applied to a completely new (unseen and drastically different) video footage. Therefore, trained models are also tested on Volan007 (drone video) and Volan008 (helicopter video) that were not previously seen by the models. It can be seen in Table 7 that models pre-trained on VOC and trained on the balanced data tend to perform better than other models. For example, for drone footage, Model 2 (trained on D-B-VOC) performs best (24.50% mAP) on Volan007 video whereas for helicopter footage, Model 6 (trained on H-B-VOC) performs best (13.88% mAP) on Volan008 video.

Fig. 9 displays the precision-recall curves for the best performing model (i.e., Model 2) tested on different test subsets (listed in Table 6). This model is trained and validated on balanced drone videos (i.e., D-B-
When tested on balanced drone (D-B) and unbalanced drone (D-U) test subsets, the model’s performance is 65.30% and 40.00% mAP, respectively. Looking at each class individually, the detection of debris has the highest AP of 81.77% when tested on D-B and 78.53% when tested on D-U. Similarly, the detection of damaged roof has an AP of 53.27% when tested on D-B and 51.89% when tested on D-U. 

Fig. 9. Precision-recall curves for best performing model (Model 2) tested on drone balanced, drone unbalanced, helicopter balanced, helicopter unbalanced, Volan007, and Volan008 subsets.
on D-U. Moreover, as shown in this Figure, the overall mAP of Model 2 when tested on Volan007 (unseen drone video) is 24.50%, with vegetation detection achieving the highest AP of 67.59%, followed by undamaged roof detection at 32.81%, and debris detection at 17.58%. However, the model struggles to detect car resulting in the lowest AP of 1.82%. In contrast, the overall mAP of Model 2 when tested on Volan008 (unseen helicopter video) is only 6.40%, supporting the argument that the camera altitude and properties could affect the accuracy of GOI detection. These results also suggest that certain GOIs (i.e., vegetation, undamaged roof, and debris) are predictable by training the CNN model on videos of previous events, and testing it on unseen footage from new events. To understand the effect of viewpoint altitude, data balancing, and pre-trained dataset on model performance, a statistical analysis is presented in the following Subsection.

5.3. Analysis of model performance for viewpoint altitude, data balance, and pre-trained weights

The effect of three key factors, namely the viewpoint altitude (low for drone vs. high for helicopter), data balancing (balanced vs. unbalanced), and pre-trained dataset (i.e., COCO vs. VOC) on model performance is investigated in this Section. Table 8 summarizes the results of statistical analysis performed on all eight models with respect to viewpoint altitude. For each trained model, a comparison is made between the performance of that model, when tested on Volan007 (drone) and VOLAN008 (helicopter) videos. For each comparison listed in Table 8, the general hypothesis is that training and testing on videos captured from the relatively same altitude yields better results. In other words, a model trained on D (H) dataset tends to show a statistically better performance when tested on D (H) dataset. Mathematically, this can be expressed by the null hypothesis \( H_0 \) that that there is no difference in the mAP between the results when the model is tested on Volan007 and Volan008, i.e., \( u_{007} = u_{008} \). The alternative hypothesis or \( H_A \) is that \( u_{007} > u_{008} \) (for models trained on drone video, i.e., Models 1, 2, and 3) and \( u_{008} > u_{007} \) (for models trained on drone video, i.e., Models 4, 5 and 6). For each comparison, the confidence level is set at 99%, test data sample is produced by randomly selecting two-thirds of the entire frames in each video (Volan007 or Volan008), and mAP is measured and averaged over 50 iterations. Next, a two-sample one-tail t-test is run to evaluate the significance for each pair. Considering the first row of Table 8 for instance, results show that Model 1 (trained on D) performs with an average mAP of 18.20% when tested on Volan007 (captured by drone), and an average mAP of 11.74% when tested on Volan008 (captured by helicopter), and this difference in mAP is statistically significant at 99% confidence level. The same trend is observed for all eight models (training and testing on video taken from relatively same altitude yields better results). However, cross training and testing, i.e., model trained on D dataset and tested on H dataset or vice versa, leads to very low accuracy. Therefore, it can be concluded that the viewpoint altitude is a critical factor that must be considered when building and deploying CNN models for GOI detection in aerial videos.

We further compare the performance of all eight models from the perspective of data balancing. As shown in Table 9, data balancing (a.k.a. under-sampling, as explained in Section 4.3) excludes a significant number of relatively less useful video frames from the training dataset, resulting in a shorter training time by a factor of 14 to 17. In particular, when pre-trained on COCO [29], reducing the size of training dataset affects slightly on performance (1.5% increase for D and 0.23% decrease for H subset). On the other hand, by pre-training on VOC [24], data balancing not only does decrease the training time, but also improves the performance by 11.89% and 3.54% for D and H subsets, respectively. In this analysis, training time is based on Intel Xeon E5-2680 v4 2.40 GHz 14-core CPU, 128 GB RAM, and NVIDIA K80 (12 GB) GPU [44]. Overall, it is observed that models trained on balanced training subset with pre-trained weights from VOC [24] (i.e., Model 2 from the D subset, and Model 6 from the H subset) outperform other models. Sample GOI detection in video frames obtained from Models 2 and 6 are illustrated in Fig. 10.

6. Summary and conclusions

The goal of this study was to facilitate natural disaster damage detection and quantification in aerial imagery using CNN models trained on footage from past disasters. To achieve this, a list of valuable GOIs was first generated by analyzing literature pertinent to disaster response and recovery. Next, in order to train and test CNN models, an in-house dataset named Volan2018 was created (using semi-supervised web mining on YouTube videos) and annotated with tags corresponding to previously identified GOIs. Volan2018 contains eight videos from hurricanes Harvey (Texas), Irma (Puerto Rico), Maria (Florida), and Michael (Florida). A total of eight CNN models were then trained, validated, and tested on different combinations of viewpoint altitude (low

<table>
<thead>
<tr>
<th>Model</th>
<th>Train data</th>
<th>Test data</th>
<th>Pre-trained on</th>
<th>Data balance</th>
<th># Training frames</th>
<th>mAP (%)</th>
<th>Training time (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D</td>
<td>Volan007</td>
<td>COCO</td>
<td>B</td>
<td>1,665</td>
<td>18.17</td>
<td>6.93</td>
</tr>
<tr>
<td>3</td>
<td>D</td>
<td>Volan007</td>
<td>COCO</td>
<td>U</td>
<td>25,137</td>
<td>16.67</td>
<td>121.79</td>
</tr>
<tr>
<td>2</td>
<td>D</td>
<td>Volan007</td>
<td>VOC</td>
<td>B</td>
<td>1,665</td>
<td>24.50</td>
<td>7.67</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>Volan007</td>
<td>VOC</td>
<td>U</td>
<td>25,137</td>
<td>12.57</td>
<td>118.96</td>
</tr>
<tr>
<td>5</td>
<td>H</td>
<td>Volan008</td>
<td>COCO</td>
<td>B</td>
<td>934</td>
<td>12.07</td>
<td>2.63</td>
</tr>
<tr>
<td>7</td>
<td>H</td>
<td>Volan008</td>
<td>COCO</td>
<td>U</td>
<td>10,297</td>
<td>12.30</td>
<td>39.04</td>
</tr>
<tr>
<td>6</td>
<td>H</td>
<td>Volan008</td>
<td>VOC</td>
<td>B</td>
<td>934</td>
<td>13.88</td>
<td>2.52</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>Volan008</td>
<td>VOC</td>
<td>U</td>
<td>10,297</td>
<td>10.43</td>
<td>37.33</td>
</tr>
</tbody>
</table>
for drone vs. high for helicopter), data balancing (balanced vs. unbalanced), and pre-trained dataset (i.e., COCO vs. VOC) to investigate the effect of these parameters on model performance. Analysis showed that CNN models trained and tested on videos captured from the relatively same altitude yields best results, while cross training and testing leads to very low accuracy. The paper also presented a novel method for balancing large video datasets using under-sampling. This approach not only did reduce the training time but also improved the performance of several models. Particularly, models pre-trained on VOC and re-trained on balanced data yielded the best results for both D and H subsets. Training and testing on Volan001 through 006 videos resulted in mAP of 74.48% (Model 3 trained on D-U-COCO and tested on D-U) and 80.69% (Model 7 trained on H-U-COCO and tested on H-B), respectively, which shows the potential of the presented technique for detection of ground assets with high fidelity in aerial imagery. While testing on completely different (unseen) footage (captured by different cameras, in different events and location), an mAP as high as 24.50% (across all classes) was achieved, with the model detecting vegetation and undamaged roof classes with ~68% and ~33% average precision, respectively.

In addition to hurricanes, we plan to investigate the capability of CNN models for GOI detection in other types of natural disasters such as tornados, typhoons, earthquakes, and wildfires, as well as analyze parameters that may influence model performance in such scenarios. While the CNN models developed in this research are based on the YOLO architecture (due to its speed and accuracy for real-time field assessment and decision-making), other CNN-based deep learning models that can achieve equal or better output with respect to speed, accuracy, and level of detail (e.g., pixel segmentation) will be also investigated in the future. Finally, Volan2018 dataset is annotated with bounding boxes which can constraint the ability of the trained CNN models to predict non-rectangular boxes (i.e., grid structure). This approach, however, could be improved particularly for detection of objects with irregular shapes and sizes (e.g., flood area, vegetation). To this end, our next revision to the dataset will contain semantic segmentation where each pixel of the image will be labeled. We expect that these improvements coupled with increased size and diversity of training data help achieve better results. Findings of this work are ultimately sought to be a key component of information exchange platforms and decision support system (DSS) applications that integrate and share different data modalities with all stakeholders involved in disaster management, ranging from ordinary people to first responders, law enforcement, local jurisdictions, insurance companies, and non-governmental organizations (NGOs).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Model training was performed on Texas A&M University’s High Power Research Computing (HPRC) clusters. The authors gratefully acknowledge the HPRC for providing necessary computational resources. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily represent those of the HPRC.

References


Fig. 10. Detection samples of Model 2 and Model 6 on Volan2018 dataset.