Automated Satellite-based Assessment of Hurricane Impacts on Roadways

Michele Gazzea, *Member, IEEE*, Alican Karaer, Mahyar Ghorbanzadeh, Nozhan Balafkan, Tarek Abichou, Eren Erman Ozguven and Reza Arghandeh, *Senior Member, IEEE*

***Abstract*—During extreme weather events like hurricanes, trees can cause significant challenges for the local communities with roadway closures or power outages. Local responders must act quickly with information regarding the extent and severity of hurricane damage to better manage recovery procedures following natural disasters. This paper proposes an approach to automatically identify fallen trees on roadways using high-resolution satellite imagery before and after a hurricane. The approach detects fallen trees on roadways via a co-voting strategy of three different algorithms and tailored dissimilarity scores. The proposed method does not rely on the large manually labeled satellite image data, making it more practical than existing approaches. Our solution has been implemented and validated on an actual roadway closure dataset from Hurricane Michael in Tallahassee, Florida, in October 2018.**

***Index Terms*—Remote Sensing, Satellite Imagery, Tree Debris Detection, Post-Hurricane Assessment, Deep-learning**

I. Introduction

Damage assessment is essential after catastrophic events like hurricanes, earthquakes, or tsunamis. Hurricanes, in particular, cause significant damages in the US, which is among the five countries most hit by natural disasters with an estimated cost of $1.75 trillion for damages from 1980 to 2019 [1].

Aerial images are a valuable source of data for hurricane damage assessment [2]. However, flying helicopters and drones over damaged areas are highly prone to weather conditions. Moreover, their high operating costs are a burden on large-scale applications. In recent years, the dramatic drop in satellites' launching cost and the growing number of satellites in orbit significantly reduced the cost of high-resolution satellite imagery [3]. Commercial satellite providers can offer high-resolution images (0.3 to 0.5 pixels/meter) with a daily and sub-daily revisiting frequency for most parts of the globe.

The combination of coverage, frequency, and cost-efficiency of satellite imagery in addition to advancements in machine learning (ML) creates a paradigm shift for enhancing situational awareness in infrastructure networks [4], [5]. In the recent literature, there are examples of using machine learning for assessing impacts of tsunamis [6], earthquakes [7], and floods [8] on infrastructure networks. In the event of hurricanes, fallen trees are one of the most common causes of damage to roadways, buildings, and electric lines. A major body of literature on hurricane-induced tree failures is focused on estimating tree failure probability concerning wind data ([9], [10]). Although such models are useful for better assessing fallen trees' consequences, few works have been done to automatically detect fallen trees using satellite images and machine learning.

Recently, supervised learning methods have been proposed to detect roadway closures using satellite images, typically by using Convolutional Neural Networks (CNNs) [11], [12], [13]. The major drawback of supervised models is that they rely highly on scant and expensive manually labeled data. It is also crucial to have clean and noise-free labels to train such models. Furthermore, supervised models, in general, are less scalable, and they need to be often retrained if we want to deploy them in another area, [14].

Unsupervised methods usually are more viable options in real-world remote sensing applications [15]. In unsupervised ML image processing approaches, features are extracted automatically for image segmentation or change analysis [16], [17]. Autoencoders (AEs) are one of the most efficient algorithms for images feature extraction. They represent data in a latent (usually smaller than the original) space preserving as much relevant information as possible.

In this paper, we focus on detecting fallen trees that cause roadway closures after hurricanes. We use high-resolution satellite images taken before and after a hurricane to perform change detection analysis and automatically locate areas where trees have landed on the roadways.

The main contributions of this paper are listed as follows:

- From the **application** point of view, our proposed approach needs limited labeled satellite images for trees in normal condition and it does not need any labeled data regarding fallen trees. In practice, our approach is unsupervised in relation to fallen trees. This increases the robustness of the approach against noisy or faulty labels, which are often acquired in satellite images and remote sensing applications. Our algorithm is also more scalable since it does not rely on an extensive training data set. Moreover, it has a lightweight open-loop architecture for fast computing time.
- From the **methodology** point of view, we propose a framework consisting of three different and powerful algorithms, i.e. a segmentation model, a spectral vege-

M. Gazzea and R. Arghandeh are with the Department of Computer Science and Electrical Engineering at the Western Norway University of Applied Sciences, Bergen 5063, Norway. E-mail: (michele.gazzea@hvl.no, reza.arghandeh@hvl.no).

Alican Karaer, Mahyar Ghorbanzadeh, Tarek Abichou and Eren Erman Ozguven are with the Department of Civil & Environmental Engineering at the Florida State University, Tallahassee, Florida. E-mail: {akaraer, mghorbanzadeh, tabichou, eozguven}@fsu.edu

Nozhan Balafkan is with StormGeo AS, Bergen, Norway. E-mail: nozhan.balafkan@stormgeo.com

tation index and Variational Autoencoders. We adapted and combined them to work together to overcome the shortage of each one of these methods to detect the fallen trees. This is done using tailored dissimilarity scores for changes in satellite images and a heuristic approach for thresholding.

II. Study Area

Hurricane Michael was one of the strongest storms hitting the Southeast coast of the United States. It made landfall as an unprecedented Category 5 hurricane in the Florida Panhandle region with maximum sustained wind speeds of 140 knots (161 mph) bringing catastrophic storm surge to the Florida State and Big Bend areas (especially Mexico Beach and Panama City) [18]. It hit the City of Tallahassee between October 10^{th} and 11^{th} 2018, leaving 1.2 million electricity customers without power in several east coasts and southern states. Estimated damage from Michael throughout the United States reached $25 billion [19]. Hurricane Michael hit the City of Tallahassee, Florida's capital, on October 10^{th} 2018. As a medium-sized city, Tallahassee has a population of 193,551 as of the year 2018 [20]. We acquired two high-resolution satellite images on a large portion of Tallahassee (Fig. 1) before and after Hurricane Michael. Images are provided by WorldView satellites, as described in Table I.

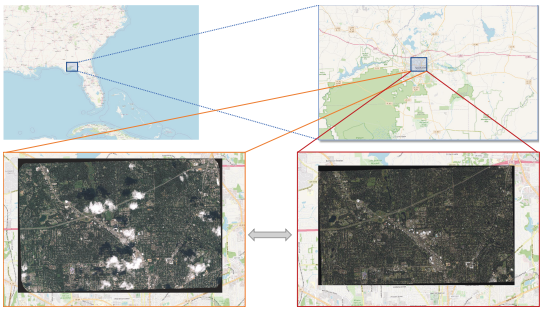


Fig. 1: Study area in Tallahassee (Florida) where satellite images before and after Hurricane Michael are acquired.

TABLE I: Satellite images used in this paper

	Date	Channels	Resolution
Pre-event image	September 14 2018	RGB-NIR	0.5m/pixel
Post-event image	October 13 2018	RGB-NIR	0.3m/pixel

For validation purposes, we use the dataset related to tree debris locations acquired from the contractual company hired by the City of Tallahassee to remove the debris.

III. Methodology

In this paper, our goal is to automatically detect fallen trees along roadways. Our method works with very limited training data and doesn't need explicit training to recognize tree debris. To do so, our major assumption is that a fallen tree will exhibit a change that is possible to detect between satellite images from the same area taken before and after a hurricane. In literature, autoencoders have been used to extract and compare important features from images and perform change detection [21]. The fundamental challenge with autoencoders is the lack of control over where the features are mapped and this is challenging in satellite images where feature vectors shall be compared. Therefore, we used Variational Convolutional Autoencoders (VAEs) [22] to overcome this issue. Relying solely on either autoencoders or variational autoencoders can be problematic due to the lack of semantic information about the changes. For example a passing vehicle or other environmental artifacts can be detected as changes but erroneously marked as tree debris. To solve the issue, we assume that a fallen tree should lead also to a change in the vegetation characteristics (coverage, chlorophyll index) in a particular location. Therefore we add semantic vegetation-related information. Given these considerations, from a practical point of view, we detect tree debris along roadways in satellite images using three different algorithms with their own limitations which we aim to overcome using them in a collaborative framework. Our selected methods are:

- AEs/VAEs: They are powerful feature extractors, and it is possible to compare the generated features in the latent space by analyzing satellite images before and after a hurricane. On the other hand, they are unsupervised, and they do not have semantic knowledge of the objects in satellite images.
- NDVI: It is one of the most common indexes for detecting vegetation in remote sensing applications. It uses the infrared band for evaluating chlorophyll-rich vegetation. However, it can't properly distinguish grown trees from other types of vegetation, for example, grass or bushes.
- Unet: It is a modern segmentation method that is possible to specifically train to recognize trees in images. However, it is susceptible to shadows and occlusions.

We extract several patches, both from the image before the hurricane (I_{bfr}) and from the image after the hurricane (I_{aft}) to cover the whole roadways in the area. Based on the assumptions above, we develop three sub-models (called hereafter blocks), each computing a tailored, customized dissimilarity score as shown in Fig. 2. We threshold the dissimilarity score histograms to flag the presence of a potential fallen tree. We combine the three blocks together using a co-voting strategy to overcome each block's inherent limitations and increase performances. The following subsections explain each block of our proposed algorithm.

A. Tree Segmentation Block

The first block is a tree segmentation algorithm. Given an RGB image I as input, the corresponding output M^{tree} is a single-channel image where trees are detected. The Tree Segmentation Block is the only supervised component of our proposed framework. To perform the tree segmentation task, we created a labeled dataset to train the model. However, it is worthwhile mentioning that the training task for tree segmentation in a given area can be done just once using a satellite image in normal conditions before the hurricane. From a practical point of view, it saves the effort and time for labeling images of fallen trees after hurricanes which would be required for traditional supervised methods.

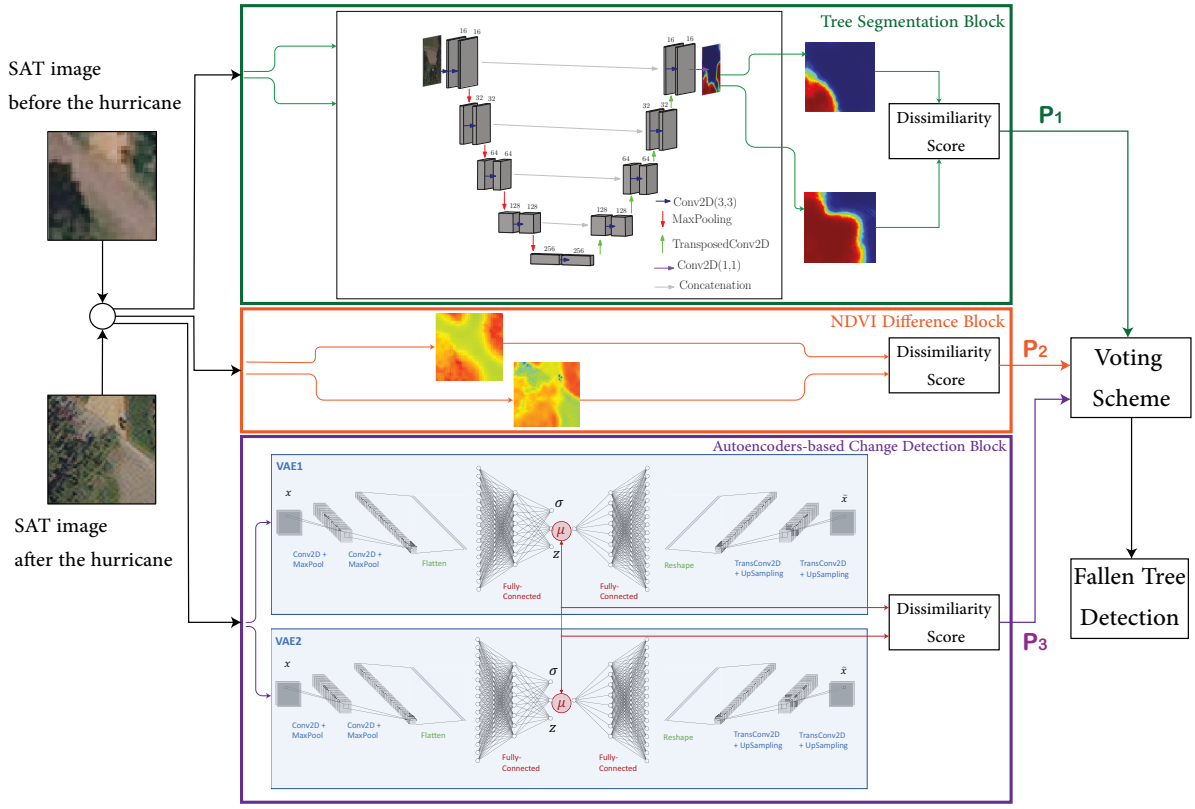


Fig. 2: Overview of the proposed approach for tree debris detection. Each of the three blocks computes a tailored, customized dissimilarity score from patches extracted before and after the hurricane. It does this using tree semantic segmentation (first block), vegetation index comparison (second block) and features comparison (third block).

We use a U-net architecture [23] which is a very popular segmentation model. Initially developed to segment biomedical images, it was successfully applied to a wide range of applications. The architecture is composed by a cascade of $[16, 32, 64, 128, 256]$ convolutional layers activated by an Exponential Linear Unit (Elu) function, followed by a batch normalization layer and a Max Polling layer, as shown in Fig. 2. During the up-sampling procedure, concatenations layer have been used to ensure better spatial localization. We use the U-net to segment each patch I_{bfr} and I_{aft} separately. Finally, to quantify a dissimilarity between the two predicted segmentation maps, a dissimilarity score $\mathcal{D}_{tree}$ is computed as the integral over the patch P of the pixel-wise difference between the segmentation map after the hurricane (M^{tree}_{aft}) and the segmentation map before the hurricane (M^{tree}_{bfr}). Such difference is then multiplied by a Gaussian kernel K centered at the center of the window (Equation 1).

$$\mathcal{D}_{tree} = \iint_{Patch} \left(M^{tree}_{aft} - M^{tree}_{bfr} \right) \otimes K \qquad (1)$$

The Gaussian kernel is a function centered at the center of the acquired patch. It has larger values in the middle and decaying values as we move towards the edges of the patch. Since the patch is acquired at the center of the roadway, such weighting function is introduced so that differences in pixel values close to the center of the patch are weighted more than differences near the edges as we are more interested in assessing the condition at the center of the window. From Equation (1) we note that the more M_{bfr} and M_{aft} are similar, the lower the value of $\mathcal{D}_{tree}$ is.

B. NDVI Difference Block

The second block takes advantage of the multi-spectral data coming from satellites to compute the Normalized Difference Vegetation Index to recognize vegetation. NDVI is a popular index in remote sensing for vegetation detection and it is defined as:

$$NDVI = \frac{\rho_{nir} - \rho_{red}}{\rho_{nir} + \rho_{red}} \qquad (2)$$

where ρ_{red} and ρ_{nir} stand for the spectral reflectance measurements acquired in the red (visible) and near-infrared regions, respectively. The reflectance measurements come directly with the optical images and are provided by the satellite operator as a bundle product. Green living plants look brighter in the near-infrared band due to chlorophyll near-infrared high reflectance. Similar to the previous block, we define a dissimilarity score $\mathcal{D}_{NDVI}$ as the integral over the patch of the difference between $NDVI$ computed after the hurricane ($NDVI_{aft}$) and the $NDVI$ computed before ($NDVI_{bfr}$), with a multiplication with a Gaussian kernel K:

$$\mathcal{D}_{NDVI} = \iint_{Patch} \left(NDVI_{aft} - NDVI_{bfr} \right) \otimes K \qquad (3)$$

C. Variational Autoencoders-based Change Detection Block

The third block is an unsupervised change detector that computes a dissimilarity value between images before and

after to detect whether a significant changes occurred. A direct pixel-wise comparison in the images is not optimal due to possibly different illumination and noise. Therefore, we use VAEs as deep feature extractors with a Siamese configuration [24]. The use of convolutional autoencoders has been motivated by the strong capabilities of deep learning methods to learn useful features in a low-dimensionality space. Variational autoencoders are also used to assure a well-defined topology of the latent space in which images are projected. This can be extremely irregular using standard autoencoders.

A variational autoencoder learns to map an input x into a distribution $p(z|x)$. It is composed by an encoder q_θ, a decoder p_ϕ, a sampler and a loss function. Since the encoder and the decoder are implemented as neural networks, θ and ϕ are all the trainable parameters of such networks. Mathematically it can be formulated as:

$$\text{Encoder:} \quad x \rightarrow q_\sigma(z|x) \quad \textit{Feature space} \tag{4}$$

$$\text{Sampler:} \quad z \sim p(z|x) \tag{5}$$

$$\text{Decoder:} \quad z \rightarrow p_\phi(x|z) = \tilde{x} \tag{6}$$

Since the distribution $P(z|x)$ is not known, Bayesian variational inference is used. We assume that the latent space, whose dimension K is chosen as hyper-parameter, follows a prior distribution which is assumed to be Gaussian in our study: $z \sim P(z) \sim N(\mu_k, \sigma_k)$, with $k = 1, \ldots, K$. We use as a loss function the sum of a reconstruction term (typical of standard autoencoders) with a regularization term calculated as the Kullback-Leibler divergence between the latent space distribution and the prior distribution as follows:

$$L(\theta, \phi) = \underbrace{MSE(x, \tilde{x})}_{\text{Recostruction loss term}} + \underbrace{\mathcal{KL}\Big(q_\theta(z|x)||p(z)\Big)}_{\text{Regularization term}} \tag{7}$$

The first term is the reconstruction loss defined as Mean Squared Error (MSE) between the original input and the reconstructed one, this term encourages the decoder to learn to reconstruct the data well from the feature space. The second term is the Kullback-Leibler (KL) divergence between the distribution of the latent space retrieved by the encoder $q_\theta(z|x)$ and the prior distribution $p(z)$. The Gaussian prior distribution $p(z)$ keeps the representations z of each input sufficiently diverse and allows a closed-form expression for the KL divergence [22]. Using the loss function in Eq. (7), one autoencoder (VAE_1) is trained to learn features from the images *before* while the other one (VAE_2) is trained with the images *after* (see the last block of Fig. 2). Features are then extracted from the bottleneck layer. The two autoencoders, VAE_1 and VAE_2, have been trained with the architecture described in Table II. We use the same idea as the previous block to calculate a dissimilarity score $\mathcal{D}_{VAE}$ between the two images (before and after the hurricane). Hence, we compute the difference between features extracted from VAE_1 and VAE_2 which is then evaluated using an *Isolation Forest* algorithm [25]. Isolation forest is a tree-structure based unsupervised learning algorithm for anomaly detection, enabling isolating anomalous points in a dataset.

From Table II, each input image is encoded into a vector $\mu \in \mathbb{R}^K$ where $K = 128$ is the dimension of latent space.

TABLE II: Autoencoder architecture

Encoder	**Decoder**
Input(80,80,3)	Dense(20*20*64)
Conv2D (3,3,32) + Relu	Reshape(20,20,64)
MaxPooling(2)	TransConv2D (3,3,64) + Relu
Conv2D (3,3,64) + Relu	UpSampling(2)
MaxPooling(2)	TransConv2D (3,3,32) + Relu
Flatten(20*20*64)	UpSampling(2)
Dense(1024)	TransConv2D (3,3,3) + Sigmoid
Dense($dim(z) = 128$)	*Output(80,80,3)*

The 3-layers architecture of our deep learning, composed by *Conv-Relu-BatchNormalization*, with a number of filters increasing by the power of 2 is a popular choice when building Convolutional Networks for computer vision tasks. On the other hand, the parameter K, which is the dimension of the latent space, is chosen empirically. The parameter K sets the amount of "compression" we are expecting from the autoencoder. A value too large does not provide a good feature representation in the latent space. On the other hand, a value too small will destroy most of the information during the encoding step. Section IV will justify our choice of the latent space dimension K.

D. Fallen Tree Detection

Our proposed framework includes three blocks, including tree segmentation, NDVI, and deep features comparison as presented in subsections III-A, III-B, and III-C. Each block computes a dissimilarity score between a pair of patches from two satellite images captured before and after a hurricane.

We extract N patches from the images before and after a hurricane to cover the roadways area. Then, we calculate the dissimilarity values per each patch location $i = 1, \ldots, N$ along the roadways for each block of the algorithm. Then, the block's outcome (a vector with N component) is represented in a histogram. We expect the histogram to be an unimodal distribution where one set (clear, debris-free locations) dominates the histogram with respect to the secondary set (obstructed locations). We used the maximum deviation method [26], especially designed for unimodal histograms, to compute a threshold and to divide the histogram in two parts. The threshold Th is selected at the point of the histogram furthest from the straight line connecting the histogram peak and the tail (Fig. 3).

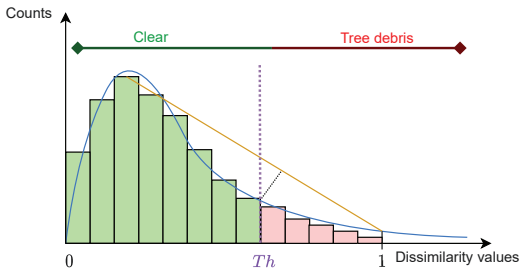


Fig. 3: Histogram of the dissimilarity values computed for each block from extracted patches. We say that there is a tree debris in a location if the corresponding dissimilarity score is greater than the threshold Th.

We also use a piece-wise linear mapping to map each dissimilarity score, computed at location $i = 1, \ldots, N$ to create

a probabilistic value of having a tree debris $P_{debris} \in (0, 1)$ in a location. The threshold Th calculated in the previous subsection corresponds to $P_{debris} = 0.5$ as shown in Fig. 4

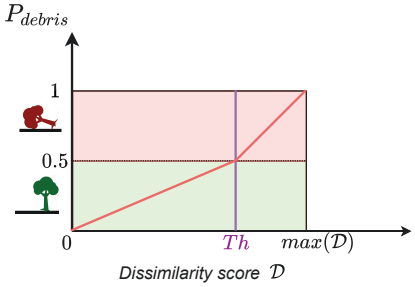


Fig. 4: Map from the dissimilarity score to the probability of tree debris P_{debris}.

Locations with $P_{debris} > 0.5$ will be flagged as fallen tree or tree debris, and clear otherwise.

E. Co-Voting Scheme

Since each block's output is independent from the other blocks, a combination strategy can possibly increase the performances. We call $P_i, i = 1, 2, 3$ the tree debris probability coming from block 1, 2 and 3 respectively. The P_{debris} is the overall tree debris probability. Here we present three possible combinations:

- *Combination 1 - Aggressive*: a location is marked as *tree debris* if at least one of the three blocks has $P_i > 0.5$.

$$P_{debris} = (P_1 > 0.5) \vee (P_2 > 0.5) \vee (P_3 > 0.5)$$

- *Combination 2 - Moderate*: a location is marked as *tree debris* if at least two of the three blocks has $P_i > 0.5$.

$$\begin{aligned} P_{debris} = (P_1 > 0.5 \wedge P_2 > 0.5) \vee \\ (P_1 > 0.5 \wedge P_3 > 0.5) \vee \\ (P_2 > 0.5 \wedge P_3 > 0.5) \end{aligned}$$

- *Combination 3 - Conservative*: a location is marked as *tree debris* if all the three blocks has $P_i > 0.5$.

$$P_{debris} = (P_1 > 0.5) \wedge (P_2 > 0.5) \wedge (P_3 > 0.5)$$

Furthermore, we can combine probabilities P_1, P_2, P_3 together using weights $\kappa_1, \kappa_2, \kappa_3$ as follows:

$$P_{debris} = \kappa_1 P_1 + \kappa_2 P_2 + \kappa_3 P_3 \tag{8}$$

where $\sum_i \kappa_i = 1$ In this way it is possible to estimate the contribution of the three blocks to the overall detection performance, and determine the weight of each block for an optimal combined decision.

IV. Results & Discussions

For validation purposes, two satellite images have been acquired for a portion of the City of Tallahassee (Florida), as described in Section II. We use a 24×24 meters sliding window (corresponding to 80×80 pixels given the satellite resolution) along the roadway's path to extract patches from the two different satellite images before and after the hurricane. Patches are spatially spaced no more than 10 meters along the road center lines. In this way, $N = 5116$ single patches are extracted, covering the whole roadway network in the study area. Due to the different resolution, the patches extracted before the hurricane have been up-sampled to match with the number of pixels of the same patches extracted after the hurricane. The three blocks have been implemented as described in subsections III-A, III-B and III-C.

The code has been implemented in Python using the sklearn library for machine learning and Tensorflow/Keras for the deep learning part. The testing platform is a computer equipped with a $10^{th} Gen$ i7 CPU, 32GB of RAM and an NVIDIA GeForce RTX 2080 Super as GPU. The images are encoded in GeoTiff format, so all pixels can be geo-referenced and precisely located in real-world coordinates. We use QGIS to integrate and visualize all the data.

To train the tree segmentation block (see section III-A), we created a labeled dataset from a part of the study area not overlapping with the locations along the roadways where we implemented and tested our algorithm. This training dataset is made of 1200 images of 160x160 pixels. The segmentation model is trained using the training dataset. The Adam optimizer has been used with a learning rate of 0.001 and exponential decays factor of 0.9. The model, with 1,944,049 trainable parameters, has been trained using binary cross-entropy as loss function. Once the model is trained to effectively recognize trees, it is used to segment trees along the 5116 patches extracted along the roadways to compute the dissimilarity scores. It is worthwhile mentioning that the model has not seen those 5116 patches during the training phase.

The variational autoencoder has been trained using the same optimizer as the tree segmentation model. The loss function is defined in Eq. 7. The encoder has 26,497,216 trainable parameters while the decoder has 3,358,659. Since variational autoencoders are unsupervised models, the usual training/validation split is not necessary.

In Table III we show the computation time needed to train each single block and compute the corresponding dissimilarity scores for all the locations along the roadways.

TABLE III: Computation time per block for training ($T_{training}$) and for the calculation of dissimilarity scores ($T_{\mathcal{D}}$) for the entire study area.

Subroutine	$T_{training}$ (sec.)	$T_{\mathcal{D}}$ (sec.)
1: Tree Segmentation (Supervised)	135	4.2
2: NDVI (no-learning)	*not applicable*	1.6
3: VAE (Unsupervised)	113	2.1
Patches extraction along roadways		1.73

We note that the most computationally-intensive operations are the training process for the tree segmentation block and the variational autoencoders block. Despite that, the entire framework is able to scan and locate fallen trees across the entire study area, which is $\approx 6Km^2$, in less than 5 minutes.

Fig. 5 shows visually an example of a hurricane-induced vegetation change. Note that the NDVI difference and the tree segmentation difference (Fig. 5e and 5h respectively) clearly illustrate such changes.

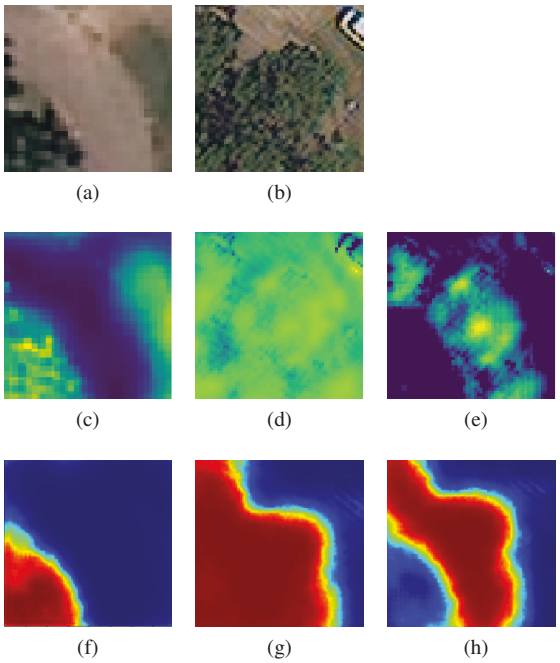


Fig. 5: An example to show a Hurricane-induced vegetation change in a patch: (a) RGB image sample acquired before the hurricane; (b) RGB image sample acquired after the hurricane; (c) NDVI of the image before the hurricane; (d) NDVI of the image after the hurricane; (e) NDVI difference; (f) tree segmentation of the image before the hurricane; (g) tree segmentation of the image after the hurricane; (h) Tree segmentation difference.

Fig. 6 shows an example of a reconstructed patch. With the proposed variational autoencoder architecture, the raw input RGB image ($3 \times 80 \times 80 = 19200$ pixels) is encoded into only 128-values vector (dimension of the latent space), thus achieving a high compression rate. Nevertheless, it is possible to decode the patch back without losing many details, as shown in Fig. 6, therefore such features catch the relevant patterns in the initial images.

Fig. 6: Example of reconstructed patch: (left) original patch extracted after the hurricane, (right) reconstructed patch.

For each patch, dissimilarity scores $\mathcal{D}_{tree}$, $\mathcal{D}_{NDVI}$ and $\mathcal{D}_{VAE}$ are calculated and normalized into the interval $(0, 1)$. To validate the algorithm, we use the geo-location of stacked piles of tree debris provided by the City of Tallahassee municipality to us, as described in Section II. The debris piles location on the map are not accurate in some cases. It can also be more than one report referring to the same fallen tree. To solve these issues, we spatially cluster the reported trees locations (using MeanShift algorithm). We introduce a distance threshold of $d = 40$ meters between the points reported by the City and the points detected by our algorithm from satellite images. The accuracy scores, True Positive (TP), False Positive (FP), and False Negative (FN), are calculated as follows and as sketched in Fig. 7.

- *True Positives:* Number of debris detected by the algorithm (red points) close to the ground-truth debris (stars)
- *False Positives:* Number of debris detected by the algorithm (red points) not close to the ground-truth debris (stars)
- *False Negative:* Number of ground-truth debris (stars) not close to debris detected by the algorithm (red points)

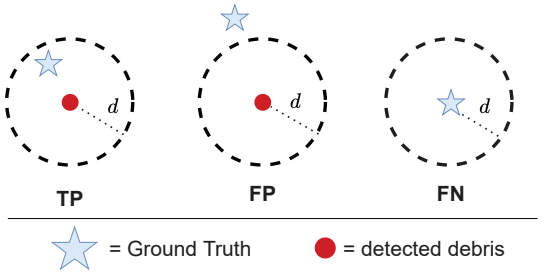


Fig. 7: Calculation of True Positive (TP), False Positive (FP) and False Negative (FN).

Then, we calculate the recall, precision and F1-score (harmonic mean of recall and precision) of the model compared with the ground truth as follows:

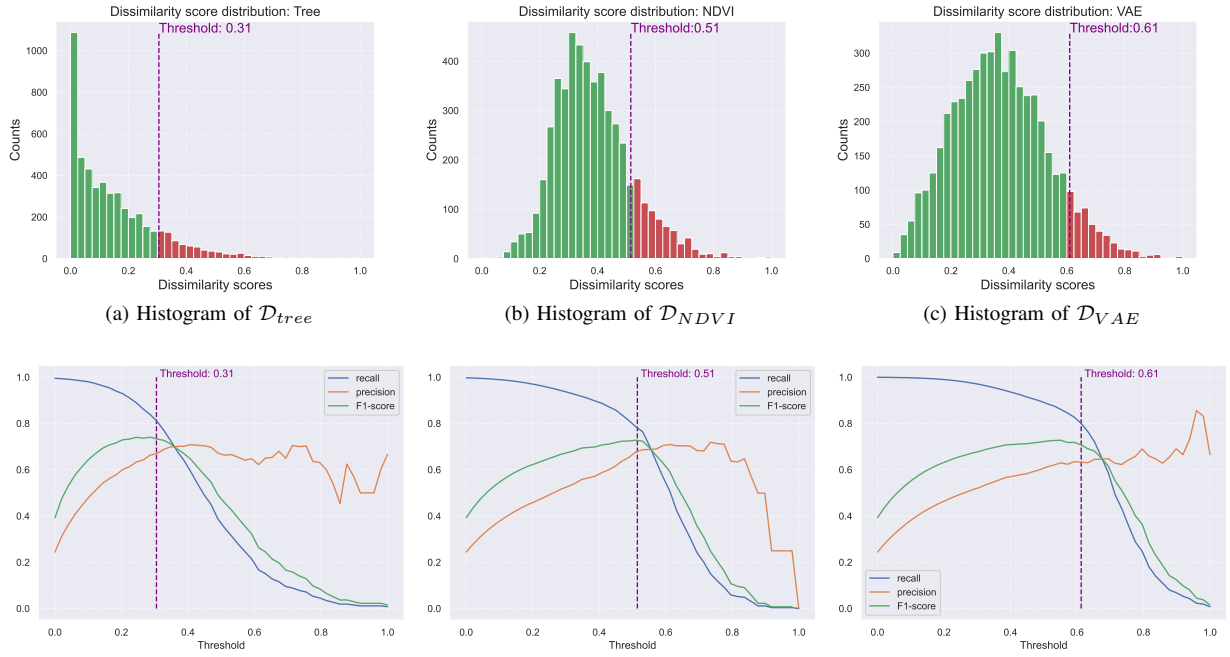
$$Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP}$$
$$F1 = \frac{2(Recall \cdot Precision)}{Recall + Precision} \tag{9}$$

Since in our application both recall and precision are important measurements, we used F1-score as a final metrics.

We first used Eqs. (9) to evaluate the performances of each block in our approach independently. To show the effectiveness of the proposed threshold Th, a sensitivity analysis is performed to evaluate the effect of the threshold on recall, precision and F1-score values. These scores are plotted as a function of the threshold in Fig. 8. We note that such an analysis is only possible when having the ground-truth. In other words, it is not possible to develop an algorithm to choose the optimal threshold that optimizes the performances in advance. We notice that the threshold we choose heuristically in the subsection III-D (vertical purple line) approximates closely the *argmax* of the F1-score. This shows the strength of our threshold choice in this case. Most importantly, it is evaluated a-priori based on the distribution of the dissimilarity scores.

Finally, we compare the different blocks as well as the different combinations proposed in Section III-E. The results are presented as bar plots in Fig. 9.

From the accuracy comparison we see that Combination 1 and 3 achieve the highest recall and precision respectively. However, Combination 1 results in an over-estimation of tree debris in the area, leading to low precision, because it merges the obstructed locations detected by all the three blocks. Combination 3 is too conservative and unable to detect most of the tree debris resulting in high number of false negatives (low recall). Combination 2 (moderate strategy) is the most balanced and achieves the highest F1-score among



(a) Histogram of $\mathcal{D}_{tree}$ (b) Histogram of $\mathcal{D}_{NDVI}$ (c) Histogram of $\mathcal{D}_{VAE}$

(d) Accuracy scores as a function of Th_{tree} (e) Accuracy scores as a function of Th_{NDVI} (f) Accuracy scores as a function of Th_{VAE}

Fig. 8: Sensitivity Analysis: first column) Tree segmentation block, second column) NDVI block, third column) VAE block. Recall, precision and F1 curves are plotted in blue, orange, green respectively as function of the threshold. The purple vertical line shows the threshold value we heuristically selected using the procedure explained in subsection III-D

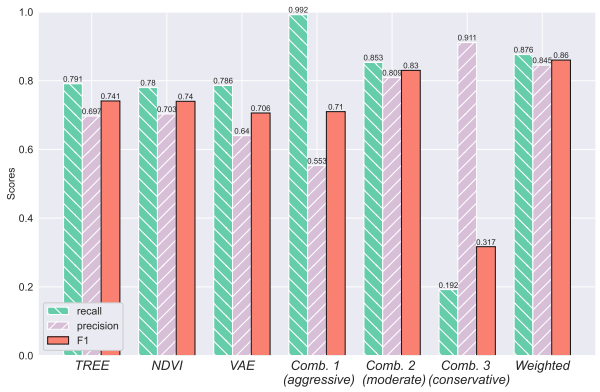


Fig. 9: Recall, precision and F1-score for each single block presented and the different combinations presented in subsection. III-E.

the three combinations. Moreover, it is possible to estimate the contribution of the three blocks to the overall detection performance, and determine the weights $\kappa_1, \kappa_2, \kappa_3$ of each block for an optimal combined decision. The weighting shows that *TREE* and *NDVI* have a slightly better F1-score than *VAE*. Therefore, we assigned proportionally-derived values to κ_1, κ_2 than κ_3 as: $\kappa_1 = \kappa_2 = 0.35, \kappa_3 = 0.3$ using the F1-score as metrics. With such weighting combination, the performances are further improved by 3.6%.

We compared our best results, with other state-of-the-art machine learning algorithms. Table IV shows the comparison results.

We noted that our approach outperforms other state-of-the-art methods. Autoencoders-based algorithms are used for change detection but they can't effectively recognize tree debris due to their fully unsupervised nature. Other machine learning-based methods exploit specific descriptions, specifically Gray-Level Co-Occurrence Matrix (GLCM) and Local Binary Patterns (LBP) to extract texture signature classified then using a Support Vector Machine (SVM). However, our approach outperform them by 13%. CNN achieves a higher recall accuracy than our method. Nevertheless, if we compare the F1-score, our model shows an improvement by 4%. It is noticeable that the CNN model is supervised, thus it has been trained directly with tree debris ground truth data to recognize them. However, the tree debris ground truth may not be available always in real-world application. Our approach shows comparable performance without relying on actual fallen trees data for training. Therefore it is more practical in areas where such labels might not be available.

TABLE IV: Comparison between our final approach and other deep-learning methods (Autoencoder-based, CNN-based) and machine learning methods (features descriptors + Support Vector Machines).

Algorithm	Recall	Precision	F1-score
Our approach	0.875	**0.845**	**0.859**
Sparse AEs [27]	0.708	0.587	0.642
Joint AEs [28]	0.841	0.668	0.744
CNN [13]	**0.947**	0.725	0.821
GLCM+SVM [29]	0.816	0.691	0.748
LBP+SVM [30]	0.828	0.698	0.757

Finally, the framework's output is used to automatically create a map showing all the detected fallen trees within the area, see Fig. 10. In case of a future hurricane, such a map can be generated quickly after the hurricane hits and used by the

city municipality or transportation authorities responsible for hurricane restoration to overview the most damaged areas. The satellite images provide wide area situational awareness after a hurricane with low cost and in a short time. Therefore, our proposed automated approach using satellite images is adding extra value to the emergency management teams on top of the information from traditional ground based inspections.

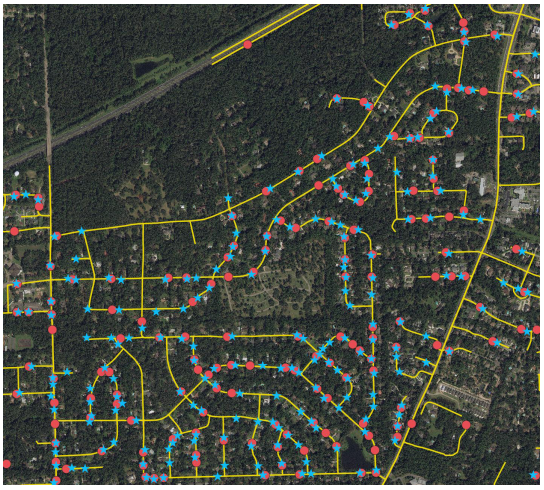


Fig. 10: Example of map showing all the detected anomalies within the area using Combination 2 in the study area. Blue stars are the reported debris locations. Red points are the debris detected by our algorithm. The yellow lines are the scanned roadways.

V. Practical Considerations for Emergency Management Teams

Satellite data has a number of advantages compared to other infrastructure monitoring approaches such as post hurricane ground-based inspection, aerial inspection with helicopters or drones, and direct reports from citizens.

Our proposed satellite based framework can serve as a complementary source for emergency management teams in addition to the traditional roadway inspection approaches especially in rural areas, remote locations, and less populated regions. Depending on the extend of hurricane damages or the weather conditions, ground-based inspection of roadways may not be possible or it is limited by roadway closures. Helicopters and drones are also prone to weather conditions. Cities can get advantage of active participation of citizens through phone calls or recently mobile applications, called city dashboards, to receive report of damages or roadway closures. However, not all cities have mobile applications in operation or they face a low adoption rate among their citizens [31], [32]. Moreover, cellular networks and internet access aftermath a natural disaster may fail. Therefore, citizens can not provide critical information to the city government using mobile applications. Above all, the tremendous drop in satellites data costs in recent years made them more cost effective than aerial inspection methods, such as helicopters and drones especially for large areas [4].

It is worth while mentioning that are some practical limitations in using optical satellite images. The first is the cloud coverage. It is not always possible to acquire a cloud-free image right after the event of hurricane. For example, in this study the first suitable image could be acquired two days after the hurricane hit.

Moreover, the accuracy of fallen tree detection depends on ecological aspects of the target area such as the tree coverage density.In our case, the city of Tallahassee in Florida is highly covered with large trees such as Tupelo (*Nyssa sylvatica*) and Red maple (*Acer rubrum*) with wide canopies that mostly make (beautiful and tourist attractive) canopy roads. In such canopy roads, it is challenging to distinguish fallen trees from standing ones. Canopy roads impose a geometric limitation for inferring fallen trees on the roads' pavement. The satellite's view from the top cannot always realize that a tree is on the pavement or if it is standing over the road. The figure below show some views of the canopy roads in Tallahassee. However,

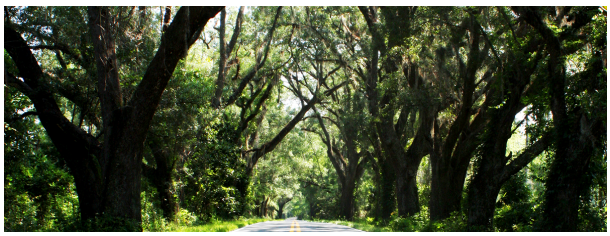


Fig. 11: A typical canopy road in Tallahassee. Photo courtesy of leontrees.org

such tangled canopy roads are limited, and their locations are already known in advance by the city municipality. They rely more on ground-based inspections or residents' reports for such areas. In our study, most false negatives occurred in such roads that brought our accuracy to 86%. In other words, the classic ground-based inspection is more critical in specific geographical areas in each city where the remote sensing approaches have limitations.

In terms of satellite data storage requirements, commercial satellite image providers and data companies offer cloud-based solutions to access and analyze the images in archive or on demand. A typical high resolution satellite image can be up to few GigaBytes. If one does not use a cloud based service, the local workstations can usually handle such images for limited studies or proof-of-concepts. There are a number of standard formats for satellite images such as GeoTiFF, NetCDF, and HDF. In this study we used GeoTiff format. A GeoTiff image is a normal bitmap image that contains additional metadata about the geo-locations of the pixels. In this way it is possible to map each pixel into a real-world coordinate system.

Finally, the proposed satellite based approach for hurricane impact study on roadways can be extended to other infrastructure networks such as electricity lines, rail roads, and natural gas pipelines. Specifically, the electric grid has a similar topology to roadways in some parts of the world like the United States where overhead power lines are laid parallel to roadways. Fallen trees induce similar threats to power lines and can cause power outages after hurricanes. Our approach with some modification can be used for electric grid damage assessment.

VI. Conclusions

This paper proposes an automated framework to detect fallen trees on roadways after hurricanes. We use two satellite images for the City of Tallahassee in Florida, acquired before and after Hurricane Michael in October 2018. Our proposed satellite-based analytical framework relies on three different methods integrated into an automated setting. Despite the challenging task from the remote sensing perspective, our solution leads to 0.86% accuracy in detecting fallen trees. Our approach is also more practical since it works with a limited training dataset and has a fast computing time. Our approach is complementary to the classic hurricane damage assessment practices. It provides the emergency management teams with a wide area of situational awareness at a lower cost and in a shorter time, which can be beneficial for stakeholders. A similar methodology can be used for other types of infrastructure networks rather than roadways. Therefore, our future work will be toward investigating different machine learning algorithms and expanding our platform to other applications such as electricity networks.

References

[1] National Oceanic and Atmospheric Administration (NOAA). Hurricane costs. https://coast.noaa.gov/states/fast-facts/hurricane-costs.html.

[2] Martin Schaefer, Richard Teeuw, Simon Day, Dimitrios Zekkos, Paul Weber, Toby Meredith, and C.J. Westen. Low-cost uav surveys of hurricane damage in dominica: automated processing with co-registration of pre-hurricane imagery for change analysis. *Natural Hazards*, 101, 04 2020.

[3] Lake Singh, William Whittecar, Marc DiPrinzio, Jonathan Herman, Matthew Ferringer, and Patrick Reed. Low cost satellite constellations for nearly continuous global coverage. *Nature Communications*, 11, 01 2020.

[4] H. Jones. The recent large reduction in space launch cost. In *48th International Conference on Environmental SystemsICES-2018-818-12 July 2018, Albuquerque, New Mexico*, 2018.

[5] M. Gazzea, M. Pacevicius, D. O. Dammann, A. Sapronova, T. M. Lunde, and R. Arghandeh. Automated power lines vegetation monitoring using high-resolution satellite imagery. *IEEE Transactions on Power Delivery*, pages 1–1, 2021.

[6] Akansha Mehrotra, Krishna Kant Singh, Madhav Ji Nigam, and Kirat Pal. Detection of tsunami-induced changes using generalized improved fuzzy radial basis function neural network. *Natural Hazards*, 77, 05 2015.

[7] M. Ji, Lanfa Liu, Runlin Du, and M. Buchroithner. A comparative study of texture and convolutional neural network features for detecting collapsed buildings after earthquakes using pre- and post-event satellite imagery. *Remote. Sens.*, 11:1202, 2019.

[8] Bo Peng, Zonglin Meng, Qunying Huang, and Caixia Wang. Patch similarity convolutional neural network for urban flood extent mapping using bi-temporal satellite multispectral imagery. *Remote Sensing*, 11(21):2492, Oct 2019.

[9] D. Gullick, A. Blackburn, D. Whyatt, and P. Vopenka. Tree risk evaluation environment for failure and limb loss (treefall) : Predicting tree failure within proximity of infrastructure on an individual tree scale. 2017.

[10] A. Kocatepe, Mehmet Baran Ulak, Grzegorz Kakareko, E. Ozguven, Sungmoon Jung, and R. Arghandeh. Measuring the accessibility of critical facilities in the presence of hurricane-related roadway closures and an approach for predicting future roadway disruptions. *Natural Hazards*, 95:615–635, 2018.

[11] Q. D. Cao and Youngjun Choe. Building damage annotation on post-hurricane satellite imagery based on convolutional neural networks. *Natural Hazards*, pages 1–20, 2020.

[12] D. E. Kislov and K. Korznikov. Automatic windthrow detection using very-high-resolution satellite imagery and deep learning. *Remote. Sens.*, 12:1145, 2020.

[13] Baolin Yang, S. Wang, Yi Zhou, Fu tao Wang, Q. Hu, Ying Chang, and Q. Zhao. Extraction of road blockage information for the jiuzhaigou earthquake based on a convolution neural network and very-high-resolution satellite images. *Earth Science Informatics*, 13:115–127, 2019.

[14] Francesco Nex, Diogo Duarte, Fabio Giulio Tonolo, and Norman Kerle. Structural building damage detection with deep learning: Assessment of a state-of-the-art cnn in operational conditions. *Remote Sensing*, 11(23):2765, Nov 2019.

[15] Wenzhong Shi, Min Zhang, Rui Zhang, Shanxiong Chen, and Zhao Zhan. Change detection based on artificial intelligence: State-of-the-art and challenges. *Remote Sensing*, 12(10):1688, May 2020.

[16] P. Zhang, M. Gong, H. Zhang, J. Liu, and Y. Ban. Unsupervised difference representation learning for detecting multiple types of changes in multitemporal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):2277–2289, 2019.

[17] E. Kalinicheva, J. Sublime, and M. Trocan. Neural network autoencoder for change detection in satellite image time series. In *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 641–642, 2018.

[18] National Oceanic and Atmospheric Administration (NOAA). Catastrophic hurricane michael strikes florida panhandle. https://www.weather.gov/tae/HurricaneMichael2018.

[19] National Oceanic and Atmospheric Administration (NOAA). Assessing the u.s. climate in 2018. https://www.ncei.noaa.gov/news/national-climate-201812.

[20] Wikipedia contributors. Tallahassee, florida — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Tallahassee,_Florida&oldid=1011330996, 2021.

[21] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1845–1849, 2017.

[22] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *CoRR*, abs/1906.02691, 2019.

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[24] Kian Ahrabian and B. BabaAli. Usage of autoencoders and siamese networks for online handwritten signature verification. *Neural Computing and Applications*, 31:9321–9334, 2018.

[25] F. T. Liu, K. M. Ting, and Z. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[26] Paul L. Rosin. Unimodal thresholding. *Pattern Recognition*, 34(11):2083–2096, 2001.

[27] E. Kalinicheva, J. Sublime, and M. Trocan. Neural network autoencoder for change detection in satellite image time series. In *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 641–642, 2018.

[28] Jérémie Sublime and E. Kalinicheva. Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the tohoku tsunami. *Remote. Sens.*, 11:1123, 2019.

[29] H. Rastiveis, E. Hosseini-Zirdoo, and F. Eslamizade. Automatic blocked roads assessment after earthquake using high resolution satellite imagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 601–605, 2015.

[30] Idrissa Coulibaly, R. Lepage, and Michèle St-Jacques. Road damage detection from high resolution satellite images based on machine learning. In *IEEE, MultiTemp 2015*, 07 2015.

[31] Jinghui (Jove) Hou, Laura Arpan, Yijie Wu, Richard Feiock, Eren Ozguven, and Reza Arghandeh. The road toward smart cities: A study of citizens' acceptance of mobile applications for city services. *Energies*, 13(10):2496, May 2020.

[32] Vaibhav S. Diwanji, Laura Arpan, Mehmet Baran Ulak, Jinghui (Jove) Hou, Eren Erman Ozguven, and Reza Arghandeh. Understanding citizens' communication channel preferences during natural disasters: A synchronicity-based, mixed-methods exploration using survey and geospatial analysis. *International Journal of Disaster Risk Reduction*, 47:101646, 2020.

Michele Gazzea, IEEE Student Member Michele Gazzea received his Bachelor's degree in Information Engineering and his Master's degree in Automation and Control Engineering, both from the University of Padova (Italy). He worked for one year in Cielle.srl in Treviso (Italy) as an R&D engineer studying and designing diagnostic techniques on milling and engraving CNC machines. He worked as a researcher for Electrolux to perform model-based analysis of power consumption in washing machines until he started his new occupation as a Ph.D. student in Western Norway University of Applied Sciences in Bergen (Norway). His research interests are in data analytics, machine learning, computer vision, and remote sensing applications.

Michele Gazzea, IEEE Student Member Michele Gazzea received his Bachelor's degree in Information Engineering and his Master's degree in Automation and Control Engineering, both from the University of Padova (Italy). He worked for one year in Cielle.srl in Treviso (Italy) as an R&D engineer studying and designing diagnostic techniques on milling and engraving CNC machines. He worked as a researcher for Electrolux to perform model-based analysis of power consumption in washing machines until he started his new occupation as a Ph.D. student in Western Norway University of Applied Sciences in Bergen (Norway). His research interests are in data analytics, machine learning, computer vision, and remote sensing applications.

Michele Gazzea, IEEE Student Member Michele Gazzea received his Bachelor's degree in Information Engineering and his Master's degree in Automation and Control Engineering, both from the University of Padova (Italy). He worked for one year in Cielle.srl in Treviso (Italy) as an R&D engineer studying and designing diagnostic techniques on milling and engraving CNC machines. He worked as a researcher for Electrolux to perform model-based analysis of power consumption in washing machines until he started his new occupation as a Ph.D. student in Western Norway University of Applied Sciences in Bergen (Norway). His research interests are in data analytics, machine learning, computer vision, and remote sensing applications.

Michele Gazzea, IEEE Student Member Michele Gazzea received his Bachelor's degree in Information Engineering and his Master's degree in Automation and Control Engineering, both from the University of Padova (Italy). He worked for one year in Cielle.srl in Treviso (Italy) as an R&D engineer studying and designing diagnostic techniques on milling and engraving CNC machines. He worked as a researcher for Electrolux to perform model-based analysis of power consumption in washing machines until he started his new occupation as a Ph.D. student in Western Norway University of Applied Sciences in Bergen (Norway). His research interests are in data analytics, machine learning, computer vision, and remote sensing applications.

Michele Gazzea, IEEE Student Member Michele Gazzea received his Bachelor's degree in Information Engineering and his Master's degree in Automation and Control Engineering, both from the University of Padova (Italy). He worked for one year in Cielle.srl in Treviso (Italy) as an R&D engineer studying and designing diagnostic techniques on milling and engraving CNC machines. He worked as a researcher for Electrolux to perform model-based analysis of power consumption in washing machines until he started his new occupation as a Ph.D. student in Western Norway University of Applied Sciences in Bergen (Norway). His research interests are in data analytics, machine learning, computer vision, and remote sensing applications.

Michele Gazzea, IEEE Student Member Michele Gazzea received his Bachelor's degree in Information Engineering and his Master's degree in Automation and Control Engineering, both from the University of Padova (Italy). He worked for one year in Cielle.srl in Treviso (Italy) as an R&D engineer studying and designing diagnostic techniques on milling and engraving CNC machines. He worked as a researcher for Electrolux to perform model-based analysis of power consumption in washing machines until he started his new occupation as a Ph.D. student in Western Norway University of Applied Sciences in Bergen (Norway). His research interests are in data analytics, machine learning, computer vision, and remote sensing applications.

Michele Gazzea, IEEE Student Member Michele Gazzea received his Bachelor's degree in Information Engineering and his Master's degree in Automation and Control Engineering, both from the University of Padova (Italy). He worked for one year in Cielle.srl in Treviso (Italy) as an R&D engineer studying and designing diagnostic techniques on milling and engraving CNC machines. He worked as a researcher for Electrolux to perform model-based analysis of power consumption in washing machines until he started his new occupation as a Ph.D. student in Western Norway University of Applied Sciences in Bergen (Norway). His research interests are in data analytics, machine learning, computer vision, and remote sensing applications.

Reza Arghandeh, IEEE Senior Member Dr. Reza Arghandeh is a Full Professor in the Department of Computing, Mathematics, and Physics and Department of Electrical Engineering at the Western Norway University of Applied Sciences (HVL), Norway. He is the Director of Collaborative Intelligent Infrastructure Lab (CI2). He is also a lead data scientist with StormGeo AS. He has been an Assistant Professor in ECE Dept, FSU, USA 2015-2018, and a postdoctoral scholar at EECS Dept, University of California, Berkeley 2013-2015. He was a power system software designer at Electrical Distribution Design Inc. in Virginia, USA, from 2011 to 2013. His research has been supported by U.S. National Science Foundation, U.S. Department of Energy, the European Space Agency, and the European Commission.