Adversarial Explanations for Informed Civilian and Environmental Protection

Theodora Anastasiou
UBITECH LTD
Limassol, Cyprus
tanastasiou@ubitech.eu

Ioannis Pastellas
UBITECH LTD
Limassol, Cyprus
ipastellas@ubitech.eu

Sophia Karagiorgou
UBITECH LTD
Limassol, Cyprus
skaragiorgou@ubitech.eu

***Abstract*—Combating crime and conditions of high physical risk in cities, the environment, and critical infrastructures requires a multifaceted approach. For sensitive problems, such as advanced situational awareness in the fields of civilian applications and environmental protection, Artificial Intelligence (AI) and Neural Network (NN) adoption has been slow due to concerns about their reliability, leading to several algorithms for explaining their decisions. Despite the possibilities for AI in critical infrastructure protection and civilian applications, many challenges still exist. For instance: (i) there are complex and high risks meaning that AI systems need to be transparent and interpretable to gain decision-maker trust; (ii) AI models may be vulnerable to imperceptible manipulations of input data even without any knowledge about the AI technique that is used; (iii) the need to efficiently process distributed, multimodal and big data coming from different, but however cheap, Internet of Things (IoT) and sensory devices (e.g., drones, cameras, accelerometers, telemetry, geomagnetic field, and proximity sensors); and (iv) many AI methods based on Machine Learning (ML) require huge amounts of training data, resulting in a Big Data computation problem. We introduce, benchmark, and demonstrate an adversarial explanations approach that we can efficiently tackle both adversarial robustness and explanation complexity of AI systems. To achieve this, we train robustified NNs and transparent explainers on big imagery data and leverage the attacks' knowledge as explanations to gain greater fidelity to the AI model. The merit of the proposed approach is that the new and robustified model has a great performance against new, unseen types of perturbations and attacks. This way, we pave the adoption of more informed and responsible AI integration in sensitive application domains.**

***Index Terms*—adversarial explanations, robust and trustworthy explainability, AI model interpretability with confidence scores**

I. Introduction

The target audience for crime prevention efforts in civilian applications is broad, including residents, Law Enforcement Agencies (LEAs), First Responders, and Government Officials. Recent developments in Artificial Intelligence (AI) have resulted in breakthroughs for many classical AI applications, such as computer vision, edge AI, and more. As a result, there are many efforts to exploit these developments for advanced situational awareness in civilian applications to safeguard the protection of people and the environment. These applications mainly serve object detection, tracking and surveillance, reconnaissance, threat evaluation, intelligence analysis, education, and training. However, despite the possibilities for AI in civilian applications, there are many challenges to consider. For instance, (i) civilian AI-systems need to be transparent and interpretable to gain decision maker trust and to facilitate real-time risk analysis of the landscape; this is a challenge since many AI-techniques are black boxes that lack sufficient transparency and interpretability of AI models preserving the internals of the decision paths producing the results, (ii) civilian AI-systems need to be robust and reliable; this is a challenge since it has been shown that AI-methods may be vulnerable to imperceptible manipulations of input data even without any knowledge about the AI-method that is used, (iii) distributed and multimodal big data are coming from different, but however cheap, Internet of Things (IoT) and sensorial devices (e.g., drones equipped with cameras, accelerometers, geomagnetic field and proximity sensors); this is a challenge since these data require different processing modalities (i.e., online vs. offline; federated vs. centralized), alignment, cleaning, and fusion to feed AI-systems, and (iv) many AI-techniques are based on Machine Learning (ML) that requires huge amounts of training data; this is challenge since there is often a lack of precise and sufficient data in civilian applications and environmental monitoring. These concerns about explainability and adversarial robustness apply to a variety of ML algorithms [1], however, in this work, we focus on the sub-field of Neural Networks (NNs) tackling the challenges of adversarial explanations to derive trustworthy and informed decisions.

State-of-the-Art (SotA) methods attempting to explain the reasoning behind NN decisions on top of imagery data focus on the generation of heatmaps or bounding boxes that indicate regions of input salient to the NN's output [2]. The problem with these explanations is that they do not report beyond a rough attention area, making it difficult to infer objective quantitative metrics and qualities about the reliability, confidence, and trustworthiness under the region of consideration. Also, most of the explanatory methods used to describe the region of consideration rely on the linearisation of a highly non-linear network, and capture relevant details only for specifically crafted or exact input [3], heavily depending on the data quality, veracity, and availability from the field of application. Adversarial attacks or the synthetic generation of adversarial training examples (i.e., adversarial examples) [4] may randomly change the NN's output. The critical issue is that the input perturbations from adversarial attacks do not align with the heatmaps or bounding boxes generated by the SotA explanation techniques. That is, without linking the robustness gained by the adversarial attacks with explanations towards an adversarial explanations approach, the NN decision has limited fidelity and validity.

Summarising, each identified attack or issue would trigger a different NN decision intervention to take the corresponding action. In this work, we combine adversarial robustness and explanation methods with a confidence score to help a decision-maker derive abundant insights and reports with high trust. We ground the NN models and adversarial explanation methods developed in promptly detecting fire incidents, as a case study for critical civilian applications applied to environmental protection. The merit of adversarial explanations for the target audience is the ability to: (i) thoroughly investigate the facts of each case in real-time with abundant evidence; (ii) allow for transparency and public scrutiny; and (iii) ensure that LEAs, First Responders, and Government Officials are informed and accountable for their actions.

We harvest imagery data collected in real-time from cameras embedded in drones participating in environmental monitoring missions and situational awareness applications to explain and reason about different aspects that may cause attacks on AI systems or harm an AI model's decision. *The merit of the application case is of high value for people's protection and environmental preservation because when a drone performs flights, it may also detect environmental crimes, including fires with augmented knowledge about adversarial attacks. The drone can then classify fires correctly with high accuracy, and promptly report back the event to a decision maker, even when adversarial attacks are present*. The contributions of this paper are, as follows:

- Harvests information from video streamed over drones equipped with high-resolution cameras for surveillance and public safety cases, environmental monitoring, and wildfire management.
- Applies a meta-algorithmic approach to improve AI systems and introduce a novel solution on adversarial explanations for situational awareness (e.g., contextually enriched and augmented data for object detection, adversarial robustness, and explanations with confidence score) to illustrate key attributes salient for classification and abundant knowledge, which is a much more reliable method of explaining an NN's decision.
- Tackles both NN's resilience and robustness, along with explanation complexity and fidelity by applying adversarial explanation methods. We have developed an extended AI pipeline to gain insights into how the AI models arrive at their conclusions, benchmark their performance, and reason with confidence scores about the causes that have resulted in their classification categories.

The rest of the paper is organized as follows: Section II provides the current literature review around innovative methods of AI for robustifying and explaining their behavior. Section III offers a detailed overview of the dataset. Section IV presents the technical architecture of the proposed AI pipeline enriched for adversarial explanations and confidence scores. Section V discusses our experimental results, and Section VI concludes the paper and provides insights to be pursued in the future.

II. Literature Review

Numerous studies have delved into innovative applications of adversarial robustness and explainability. Among the notable contributions, researchers have explored advanced adversarial robustness algorithms, XAI, and hybrid approaches combining both merits of adversarial robustness and explanatory analysis. The breadth of research underpins the importance of leveraging diverse AI technologies to mitigate AI systems risks either due to data quality issues or adversarial attacks on AI models. In the proposed work, we improve the reliability and resilience of NNs with augmented examples, informed explanations, and reported confidence scores.

Gao et al [5] study the adversarial robustness of deep neural networks for classification tasks. Through concrete classification examples and matrix-theoretic derivations, they show that the adversarial fragility of neural network-based classifiers comes from the fact that very often neural network only uses compressed features to perform the classification tasks. Thus in adversarial attacks, one needs to add perturbations to change the small subsets of features used by the neural networks. Their theoretical results show that the neural network's adversarial robustness can degrade as the input dimension d increases.

Benchama et al. [6] introduce an intrusion detection system that harnesses Generative Adversarial Networks (GANs), Multi-Scale Convolutional Neural Networks (MSCNNs), and Bidirectional Long Short-Term Memory (BiLSTM) networks, supplemented by Local Interpretable Model-Agnostic Explanations (LIME) for interpretability. They generate synthetic network traffic data, encompassing both normal and attack patterns, and feed it into an MSCNN-BiLSTM architecture for intrusion detection. The integration of LIME allows them to explain the model's decisions.

Card et al. [7] explore the adversarial vulnerabilities of a neural network-based malware classification system under the spectrum of dynamic and online analysis environments. They train a Feed Forward Neural Network (FFNN) to classify malware categories and use the state-of-the-art method, SHapley Additive exPlanations (SHAP) to inform the adversarial attackers about the features with significant importance on classification decisions. Their results demonstrate a high evasion rate for some attacks' instances, showing a clear vulnerability of a malware classifier for such attacks.

Luo et al. [8] investigate the privacy risks of Shapley value-based model interpretability methods using feature inference attacks, i.e., reconstructing the private model inputs based on their Shapley value explanations. They present two adversaries: (i) the first adversary reconstructs the private inputs by training an attack model based on an auxiliary dataset and black-box access to the model interpretability services; and (ii) the second adversary, even without any background knowledge, successfully reconstructs most of the private features by exploiting the local linear correlations between the model inputs and outputs.

Li et al. [9] study Graph Neural Network (GNN) explainers under adversarial attacks. They found that an adversary who is slightly perturbing the graph structure can ensure the GNN model makes correct predictions, but a GNN explainer may yield a drastically different explanation on the perturbed graph. They designed two methods (i.e., one is loss-based and the other is deduction-based) to realize the attack and evaluated their attacks on various GNN explainers showing that the explainers are fragile.

Woods et al. [10] extend a methodology for adversarial explanations (AE) to state-of-the-art reinforcement learning frameworks, including MuZero [11], and propose improvements to the base agent architecture. They demonstrate that this technology has two applications: (i) for intelligent decision tools; and (ii) to enhance training frameworks. In a decision support context, adversarial explanations help a user make the correct decision by highlighting those contextual factors that would need to change for a different AI-recommended decision.

Vascotto et al [12] propose a test to evaluate the robustness of non-adversarial perturbations and an ensemble approach to analyse more in-depth the robustness of XAI methods applied to neural networks and tabular datasets. They show how leveraging manifold hypothesis and ensemble approaches can be beneficial to an in-depth analysis of the AI model's robustness.

Chowdhury et al. [13] employ the notions of necessity and sufficiency from causal literature to come up with a novel explanatory technique called SHifted Adversaries using Pixel Elimination (SHAPE), which satisfies all the theoretical and mathematical criteria of being a valid explanation. They show that SHAPE is an adversarial explanation that fools causal metrics that are employed to measure the robustness and reliability of popular importance-based visual XAI methods, outperforming popular explanatory techniques like Grad-CAM and Grad-CAM++ [14].

The practical application of AI on a large scale necessitates the ability to understand and justify the predictions and decisions made. This requirement underpins the pivotal role of XAI and adversarial robustness with guarantees in fostering trust between AI systems and human actors. By providing transparency and clarity in the decision-making process, adversarial explanations bridge the gap between the complex algorithms employed by AI and the in-depth understanding of end users. This transparency cultivates trust and confidence in AI systems, thereby facilitating their widespread adoption and integration into various domains with high and safe guarantees, including civilian applications. In essence, the ability of AI to offer interpretable explanations for its decisions is paramount for realising its full potential and ensuring harmonious collaboration between humans and machines. Existing works are limited either by only addressing the adversarial robustness or the explainability aspects to derive interpretable cause-effect insights. Compared to the above-mentioned approaches, the scientific contribution of this work is the combinatorial approach toward adversarial explanations to derive more informed and interpretable decisions with objective scores. We benchmark and validate our approach in usage scenarios of situational awareness applications and environmental monitoring. The differentiation of the proposed approach from existing efforts contributes to faster root cause identification for object segmentation and detection tasks and has a direct derivative in the development of more robust and trustworthy solutions.

III. Dataset Overview

A. Fire Images

For advanced situational awareness in critical infrastructures and forest protection, fire images are a diverse collection depicting various fire incidents. These images have been sourced from multiple datasets to ensure a wide range of fire types, sizes, and environmental contexts. They include a variety of fire incidents and environmental conditions captured from different heights with different resolutions. The data sources of this work are (i) fire images captured by the drone camera of the civilian and environmental application we have developed for this study, (ii) public fire images created for the NASA Space Apps Challenge in 2018, and (iii) a publicly available Fire Detection Dataset. The fire images are characterized by the following features:

- **Diversity**: Images include indoor and outdoor fires, ranging from small flames to large conflagrations.
- **Complexity**: Scenes contain varying levels of smoke, lighting conditions, and occlusions.
- **Annotations**: Each image is annotated with bounding boxes around the fire regions, providing ground truth for AI model training and evaluation.
- **Big Data**: Both by means of volume and velocity, as large quantities of data measured in gigabytes have been used for training, while the data has been captured by a camera embedded in a drone and processed in real-time.

The total gigabyte size of harvested images is 10GB. A sample of the dataset is shown in Figure 1.

B. Adversarial Attack Images

The adversarial attack images comprise images intentionally modified to deceive the object detection module of an AI system. These adversarial examples are designed to test the robustness of the AI model under challenging conditions. Essentially, the same images have been perturbed and alternated by adversarial methods, as detailed below.

C. Dataset Statistics and Attacks

In the experiments of YOLO [15] and Grad-CAM [16] using a CNN-ResNet50-Classifier, the dataset consists of the above-mentioned fire images and fire images after they had been attacked. So in the conducted experiments, there are two

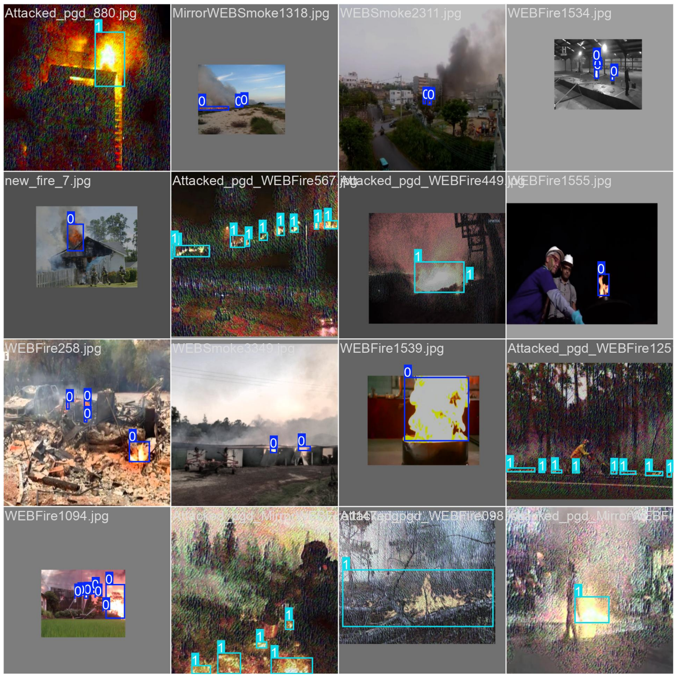

Fig. 1. Training Data (i.e., normal fire object = 0; attacked fire object = 1)
Training sample data

classes, fire (benign) and attacked_fire, and a clear discrimination between the two classes with the criteria of the attacks. The dataset consists of a balanced number of images from both categories (i.e., normal and attacked) to ensure fair evaluation. Table I provides a summary of the dataset statistics.

TABLE I
DATASET STATISTICS

Category	Training Set	Test Set
Fire Images	3527	150
Adversarial Attack Images	3527	150
Total Number	7054	300

IV. Logical Modules and Adversarial Explanations Architecture

The Adversarial Explanations Architecture and its logical modules are shown in Figure 2, and include (i) Data Collection using a camera embedded in a drone; (ii) Preprocessing and Analytics Module which is responsible for image curation, cleaning, and performs resizing, and filtering; (iii) Training and Evaluation of a CNN Model which serves as the baseline for the classifier to be used for the initialization of the Attacks; (iv) Adversarial Imagery Examples Generation through the CNN-Attacked Detector which takes as input the preprocessed images, passes them through the initialized evasion attacks using the ART library to modify them for generating adversarial imagery examples; and (v) Adversarial Examples usage; for the Benchmarking, and Validation through informed and interpretable XAI Features and Confidence Scores. Specific information necessary for reproducing the results of this study is provided in this paper regarding the datasets collected and used, the AI Models and their dimensions, the hardware specifications, and the statistical analysis. More details can be also provided upon targeted requests via email to the authors. The following sections present the logical modules in detail.

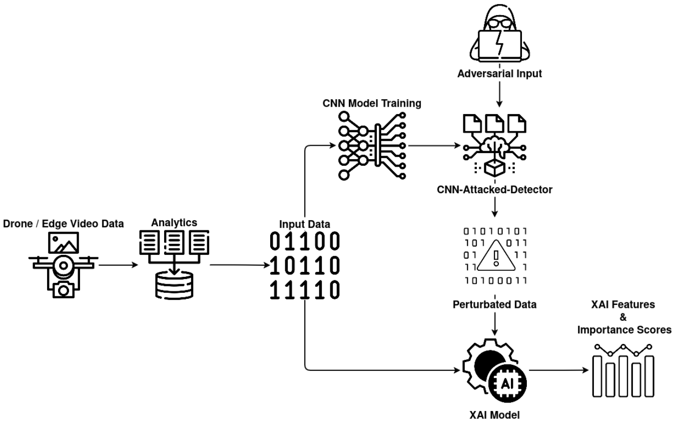

Fig. 2. Adversarial Explanations Architecture

A. Data Collection

The first logical module is about the collection of the data after being recorded and streamed over a camera embedded in a drone. After steaming the data via a streaming media server, we store the data for local processing and training in the cloud.

B. Preprocessing Module

The preprocessing module ensures that the features of the collected images are suitable for a situational awareness scenario tackling wildfires and environmental monitoring. In this step, the images are resized, reshaped, or filtered to ensure abundant feature extraction. The preprocessing module uses Keras and skimage. After performing visual data exploration, we then apply filtering, resizing, and data normalization under a specific numeric range, i.e., [0-1]. We then prepare and split the images to perform multiclass or binary classification. The multiclass classification aims to predict the image category and report if the AI system has been attacked, while the binary classification only distinguishes between attacked (i.e., malicious) and non-attacked (i.e., normal) images. The images are transformed into arrays using Keras. Next, we apply, as detailed below, filtering methods to clean and make the dataset features more visible. More specifically, we use the Simple Linear Iterative Clustering (SLIC), which is a K-Means-based image segmentation method to achieve higher model performance results. This algorithm performs K-means in the 5D space of color information and image location.

C. Training and Evaluation of a CNN Model

We then train and evaluate a CNN Model on the preprocessed data which is later used as the baseline classifier for the initialization of a given attack method.

The training and evaluation of the CNN Model facilities to build a model that serves as the baseline to investigate its robustness measured through accuracy under two concrete conditions: (i) after applying the evasion Projected Gradient Descent (PGD) attack of ART library; and (ii) after performing adversarial training to quantify at which percentage the accuracy of the CNN Model can recover from the attack as shown in V. To generate the Adversarial Imagery Examples, we use the classifier of the previously trained CNN Model to initiate informed (i.e., labeled) PGD attack. The data augmented with labels is then used as the Adversarial Imagery Examples for Experimentation and Evaluation with XAI Features and Confidence Scores. During the AI model training, we adhere to a standard methodology by partitioning the dataset into distinct training (70%) and test sets (30%) to rigorously assess the performance accuracy of our model. Furthermore, to build the CNN Model, we use different Keras layers, including Sequential, Dense, Flatten, Conv2D, MaxPooling2D, Activation, and Dropout. The activation functions are different for multiclass and binary classification according to the supported learning objective, and thus training task. More details for the configuration parameters of the CNN model can be found in Table II. The accuracy of the CNN Model is evaluated using the test set and classification reports.

D. Adversarial Imagery Examples Generation

Adversarial Machine Learning (AML) by definition, is a class of data manipulation techniques that cause alterations in the behavior of Artificial Intelligence (AI) systems while going unnoticed by humans. These alterations can cause serious vulnerabilities to mission-critical applications, such as civilian or environmental monitoring applications, attacks on autonomous vehicle navigation systems, surveillance systems, and more. While the adversarial examples are useful for augmenting benign datasets, there is an imminent need for end-to-end frameworks that enable the exploration of realistic adversarial attacks with varying threat models. For instance, in our case, we launch realistic attacks for image detection and classification tasks. This enables early adversary detection before people or public places are in danger due to attacks or other harmful situations. Attacks on the AI system may come in at different stages, however, in this scenario, we assume that the data are being modified by the adversary at the early stage of the collection or processing of the data.

1) Evasion Attacks: In this study, we consider adversaries, under the notion of evasion attacks, that cause an AI system to inaccurately identify the characteristics of the data. Through the application of augmentation, perturbations, and noise, we managed to strengthen a detection system. In this context, the adversary tries to puzzle the AI system during the inference mode by manipulating the data. For instance, the adversary may exploit a vulnerability in the drone's camera and compromise the integrity of the captured data by maliciously crafting it. We proceed by further augmenting our dataset with noise and using PGD attack method to evaluate its robustness under adversarial setups and defense (e.g., TotalVarMin, JpegCompression, and SpatialSmoothing) algorithms for recovery. Benchmarking over the defense algorithms was performed, but the recovery rate did not exceed 20% and thus it was not included in our experimental evaluation. The augmented dataset results in generating adversarial imagery examples using the ART library. As previously mentioned for the initialization of the attacks, we used a pre-trained CNN Model for baseline.

The generation of the Adversarial Imagery Examples using PGD attack, is depicted in Figure 3 and Figure 4.

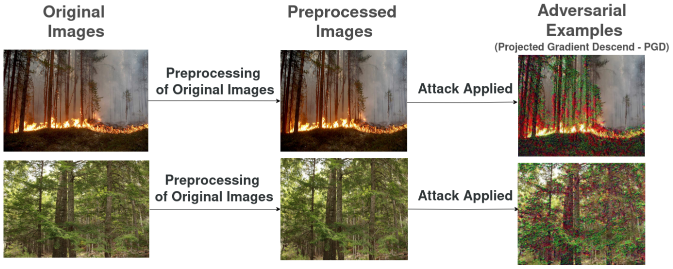
Original Images
Preprocessed Images
Adversarial Examples
(Projected Gradient Descend - PGD)
Preprocessing of Original Images
Attack Applied
Preprocessing of Original Images
Attack Applied

Fig. 3. Preprocessing Module

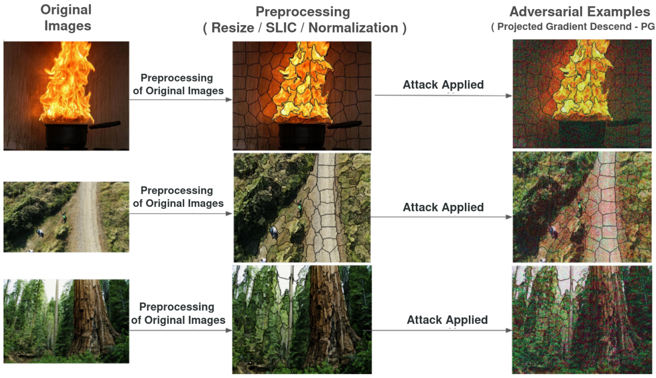
Original Images
Preprocessing
(Resize / SLIC / Normalization)
Adversarial Examples
(Projected Gradient Descend - PGD)
Preprocessing of Original Images
Attack Applied
Preprocessing of Original Images
Attack Applied
Preprocessing of Original Images
Attack Applied

Fig. 4. Preprocessing Module with SLIC filter

E. Experimentation and Evaluation through XAI Features and Confidence Scores

To benchmark the adversarial robustness with explanations, we use the adversarial imagery examples to train and evaluate a Deep Neural Network (DNN). The evaluation of the DNN model is essential for this task, to establish the performance, efficiency, and robustness of the DNN. The usage of the adversarial examples is crucial and in combination with the experimentation and evaluation through abundant XAI Features and Confidence Scores can give significant results for the performance and the robustification of the DNN Models. Last, after performing the evaluation, the DNN model both trained on labeled adversarial and non-adversarial data, is ready to harvest images and classify them correctly. Once the end user (e.g., an officer, an administrator, etc.) receives the predicted score of an image is then able to take informed actions.

In 5 the overall methodology of the paper is presented. At first level, is the adversarial generation that is used to for generating attacked images instances that will be used to complete the dataset containing both the benign and the attacked images as discussed above. Using this new dataset (benign and attacked images) YOLOv8, and a CNN-based Classifier is built, training on two classes of data, 'fire' (denoting the normal instances) and 'attacked_fire' (denoting the adverasial examples). In the CNN classifier Grad-CAM is used to get the saliency pixel importance map.

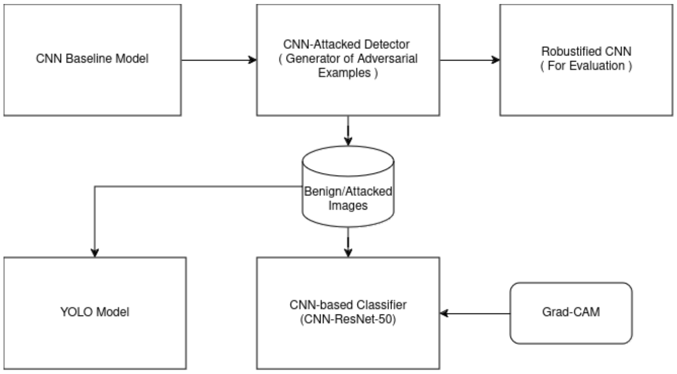
CNN Baseline Model
CNN-Attacked Detector (Generator of Adversarial Examples)
Robustified CNN (For Evaluation)
Benign/Attacked Images
YOLO Model
CNN-based Classifier (CNN-ResNet-50)
Grad-CAM

Fig. 5. Overall Methodology of this work

V. YOLO & XAI IMPLEMENTATION

The Adversarial Imagery Examples are used to investigate the behavior of XAI-DNN models (i.e., in this case, a YOLO Model, and a CNN with Grad-CAM) on a fire image detection and segmentation scenario for environmental situational awareness.

In this section, we present two approaches to extract explanations of visual data, both with YOLO bounding boxes and Grad-CAM feature importance heatmaps.

A. YOLOv8 Model

YOLOv8 [15] builds on the foundational principles of its predecessor model but introduces several architectural improvements to enhance both accuracy and efficiency. One of the key enhancements in YOLOv8 is the incorporation of a more sophisticated backbone network, designed to better capture intricate features from input images. This backbone employs a combination of residual blocks, cross-stage partial networks (CSPNet), and an enhanced path aggregation network (PANet), which both contribute to improving feature extraction and multi-scale feature fusion. These advancements enable YOLOv8, depicted in Figure 6, to handle complex detection tasks with higher precision, particularly in scenarios involving small or overlapping objects.

YOLOv8 in our case serves as a strong detector and classifier, and at the same time provides explanations through its bounding boxes prediction that surround the detected object and class, reporting a Confidence Score.

Another significant feature of YOLOv8 is its optimized head architecture, which refines the prediction process for bounding boxes, object scores, and class probabilities. The model integrates advanced techniques such as anchor-free detection and dynamic convolution, which reduce computational complexity while maintaining high detection accuracy. Additionally, YOLOv8 leverages an efficient neck structure that enhances the flow of information between the backbone and the head, further boosting performance. Furthermore, YOLOv8 is available in different sizes, i.e., nano, small, medium, large, and extra-large, allowing users to select a model that best fits their specific requirements in terms of speed, memory, and computational resources. The difference between these models is in the number of layers and neurons of each layer. Figure 6 presents an abstract architecture of the YOLOv8 model, consisting of its main components, the backbone, and the head.

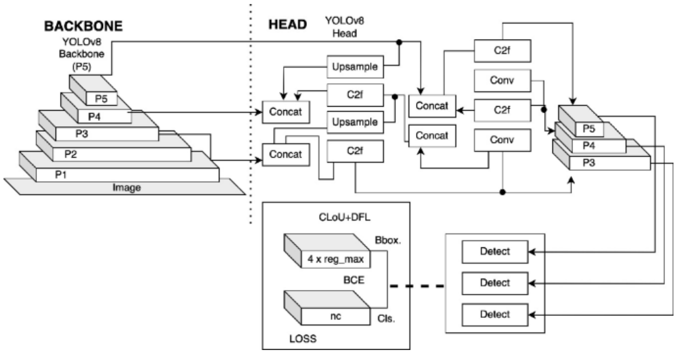

Fig. 6. YOLOv8 Architecture via Wikimedia Commons

In this scenario, the YOLO data and configuration file consists of two classes in which the YOLO is trained. The "fire" class which corresponds to the normal/benign images with fire along the coordinates of the bounding boxes surrounding fires. So there is a clear distinction between images that have normal fire and images that have been attacked and altered. The "attack fire" class corresponds to the images with fire that has been attacked along the coordinates of the bounding boxes surrounding these "attacked" fires. Thus the model receives the data mentioned above(benign fire images & attacked fire images) and can predict either fire (benign) or attacked_fire as seen in the figures 7 and 8.

B. Grad-CAM on a CNN-based Classifier

1) Grad-CAM Overview: Grad-CAM [16] is a visualization technique that highlights the regions in an input image that is most influential in CNN's decision-making process. By utilizing the gradients of the target class flowing into the final convolutional layer, Grad-CAM produces a heatmap that emphasizes the important regions for predicting the class.

2) Application to Fire Images and Adversarial Fire Classification Tasks: The primary objective of our CNN-based classifier is to accurately identify images containing fire or fire under adversarial attack. To understand the model's reasoning and robustness against such attacks, we employ Grad-CAM to visualize the decision-making process for both original fire images and adversarially attacked fire images.

3) Implementation Details: We use Grad-CAM by following the steps below:

1) Train a CNN classifier to classify between benign fire and attacked fire. The classifier builds on ResNet-50 [17] and uses an additional layer of neural network to perform the classification.
2) Pass the input images through the CNN to obtain feature maps from the last convolutional layer.
3) Compute the gradients of the target class score to these feature maps.
4) Average the gradients over all spatial locations to obtain the neuron importance weights.
5) Perform a weighted combination of forward activation maps, followed by a ReLU activation, to produce the Grad-CAM heatmap.

The resulting heatmaps are superimposed on the original images to provide intuitive visual explanations, highlighting the critical regions influencing the classifier's decisions to the detected class.

VI. Experimental Results

This section describes the experimental results of the adversarial explanations approach to derive informed decisions and ground them with a real application scenario coming from an environmental situational awareness application, i.e., a case study on wildfires. It is critical to have the capacity to evaluate the decisions of an AI Model specifically in the case of wildfires and life protection.

The experiments were conducted on local machines equipped with 15.4GB of memory and 12 CPUs, each operating at 2.60 GHz.

The preprocessing phase took approximately ˜15 minutes, while training of the CNN model - baseline classifier required around ˜40 minutes for the entire dataset. Moreover, the generation of adversarial examples consumed ˜20 minutes and finally, the adversarial training of the CNN lasted about ˜170 minutes.

The training of the YOLOv8 took ˜6 hours to achieve the reported results and the CNN-ResNet50-Classifier training took ˜2.5 hours.

TABLE II
CNN Configuration Parameters

Num of Epochs	20
Batch Size	15
Input Layer	(400, 500)
	units:8
	kernel size: (3,3)
	activation function: relu
	max pool size: (2,2)
	dropout: 0.25
Output Layer	activation function: softmax
Optimizer	adam
Crossentropy Loss	sparse categorical crossentropy

Table IV presents the configuration parameters of the CNN and YOLO models.

TABLE III
YOLO Configuration Parameters

n	parameters	layer	arguments
0	928	Conv2D	[3, 32, 3, 2]
1	18560	Conv2D	[32, 64, 3, 2]
2	29056	BottleneckCSP	[64, 64, 1, True]
3	73984	Conv2D	[64, 128, 3, 2]
4	197632	BottleneckCSP	[128, 128, 2, True]
5	295424	Conv2D	[128, 256, 3, 2]
6	788480	BottleneckCSP	[256, 256, 2, True]
7	1180672	Conv2D	[256, 512, 3, 2]
8	1838080	BottleneckCSP	[512, 512, 1, True]
9	656896	SPPF	[512, 512, 5]
10	0	Upsample	[None, 2, ’nearest’]
11	0	Concat	[1]
12	591360	BottleneckCSP	[768, 256, 1]
13	0	Upsample	[None, 2, ’nearest’]
14	0	Concat	[1]
15	148224	BottleneckCSP	[384, 128, 1]
16	147712	Conv2D	[128, 128, 3, 2]
17	0	Concat	[1]
18	493056	BottleneckCSP	[384, 256, 1]
19	590336	Conv2D	[256, 256, 3, 2]
20	0	Concat	[1]
21	1969152	BottleneckCSP	[768, 512, 1]

A. Adversarial Training Evaluation

Evasion attacks on AI / ML models are essential to build a robust system that not only can identify the class of the data but also estimate if the data is being attacked or manipulated by an external factor. Table V gives a summary of the results of the Adversarial Imagery Examples evaluation regarding the accuracy of the baseline CNN Model. We have used the K-fold cross-validation method in all experiments. This method enables the model's accuracy measurement in different data samples and avoids overfitting the data over the model before the dataset is split into different training and test sets. As is shown, the performance of the baseline CNN Model has an average of 0.87 accuracy in 5-folds. On the other hand, the PGD attack method has imposed a major influence on the model's accuracy, with a significant accuracy degradation of 0.108. In addition, as reported in V, the robustified CNN has an average of 0.864 in 5-folds, which means that adversarial training seems to be an effective way to recover the overall model robustness. In this paper we only use PGD for our experimentation, however we plan to evaluate our experiments with multiple attacks in the future.

B. YOLOv8 Evaluation

The YOLOv8 Model has been evaluated using several key metrics that measure its performance in object detection tasks. These metrics differ from the metrics in standard image classification since object detection includes the bounding box predictors besides the classes. These metrics help to quantify the accuracy, precision, and overall effectiveness of the model in identifying and localizing objects, including fire, within images. Among these metrics, mean Average Precision (mAP) is one of the most important, and is used in this work for the YOLOv8 experiments. Along these metrics, we further use standard ML metrics, such as accuracy and precision to

TABLE IV
CNN-RESNET50-CLASSIFIER CONFIGURATION PARAMETERS.

Layer Type	**Arguments**
Conv2d	(3, 64, kernel_size=7, stride=2, padding=3)
BatchNorm2d	(64)
ReLU	(inplace=True)
MaxPool2d	(kernel_size=3, stride=2, padding=1)
Layer 1 (Bottleneck Blocks)	
Conv2d	(64, 64, kernel_size=1, stride=1)
BatchNorm2d	(64)
Conv2d	(64, 64, kernel_size=3, stride=1, padding=1)
BatchNorm2d	(64)
Conv2d	(64, 256, kernel_size=1, stride=1)
BatchNorm2d	(256)
Downsample	Conv2d (64, 256, kernel_size=1, stride=1)
Layer 2 (Bottleneck Blocks)	
Conv2d	(256, 128, kernel_size=1, stride=2)
BatchNorm2d	(128)
Conv2d	(128, 128, kernel_size=3, stride=2, padding=1)
BatchNorm2d	(128)
Conv2d	(128, 512, kernel_size=1, stride=1)
BatchNorm2d	(512)
Downsample	Conv2d (256, 512, kernel_size=1, stride=2)
Layer 3 (Bottleneck Blocks)	
Conv2d	(512, 256, kernel_size=1, stride=2)
BatchNorm2d	(256)
Conv2d	(256, 256, kernel_size=3, stride=2, padding=1)
BatchNorm2d	(256)
Conv2d	(256, 1024, kernel_size=1, stride=1)
BatchNorm2d	(1024)
Downsample	Conv2d (512, 1024, kernel_size=1, stride=2)
Layer 4 (Bottleneck Blocks)	
Conv2d	(1024, 512, kernel_size=1, stride=2)
BatchNorm2d	(512)
Conv2d	(512, 512, kernel_size=3, stride=2, padding=1)
BatchNorm2d	(512)
Conv2d	(512, 2048, kernel_size=1, stride=1)
BatchNorm2d	(2048)
Downsample	Conv2d (1024, 2048, kernel_size=1, stride=2)
AdaptiveAvgPool2d	(output_size=1)
Linear	(2048, 2)
Loss	Crossentropy loss
Optimizer	adam

TABLE V
CNN ADVERSARIAL TRAINING RESULTS

	CNN Training	CNN Attacked - PGD	Robustified CNN
k-fold	Accuracy		
1	0.87	0.13	0.85
2	0.90	0.10	0.89
3	0.87	0.10	0.89
4	0.84	0.11	0.90
5	0.87	0.10	0.79
Avg 5-fold	0.87	0.108	0.864

evaluate the performance of YOLOv8, as an object detection classifier, and for the baseline CNN-based classifier where Grad-CAM was used.

1) Mean Average Precision (mAP): The mAP metric is widely used to evaluate the performance of object detection models. It combines precision and recall across different threshold levels (e.g., confidence) to provide a single performance score. Precision measures the proportion of true positive detections (correctly identified fires) out of all positive detections made by the model (including false positives). Recall measures the proportion of true positive detections out of all actual instances of the object in the dataset (including false negatives). The mAP is calculated by taking the average of the Average Precision (AP) for each class. AP is determined by plotting the precision-recall curve for each class and calculating the Area Under the Curve (AUC). In the fire detection context, a high mAP score indicates that the YOLO model effectively identifies and localizes fires and humans with few false positives and false negatives.

An important clarification on Precision and mAP, and on how we used them in this paper. Precision is calculated for both CNN and YOLO based solely if a class is detected or not. It does not take into consideration the IoU (Intersection over Union). So, in case of YOLO if the there is a detection of a class(even for one object), we count it as True Positive (TP) regardless of the predicted box coordinates.

We present some baselines, from some of the algorithms used in this work, on general benchmarks, that can be used to compare the performance of our models. Table VI shows the metrics of the Res-Net50 on CIFAR-100 dataset, and YOLO-v8 on the COCO dataset.

Table VII summarizes the metrics values used to evaluate the models. Both kind of models the `YOLOv8 large` model and the `CNN-ResNet50`, achieved an impressive accuracy of 0.95 and 1.0 respectively. Accuracy is a crucial metric that indicates the correctly predicted instances out of the total instances. A high accuracy value signifies the model's effectiveness in correctly identifying objects within the dataset. The `YOLOv8 small` model achieved an accuracy of 0.925, which is slightly lower than the large model. In terms, of accuracy, the CNN-based classifier outperforms YOLO, scoring 1.00 in accuracy.

Precision, which measures the proportion of true positive detections among the positive detections (in our case the attacked fire images), was found to be 1.00 in both YOLO and the CNN classifier. This indicates that these models had no false positives in their predictions, showcasing their reliability in terms of precision, for the attacked images.

The mAP50 value for the `YOLOv8 large` model is 0.47 and for the `YOLOv8 small` model is 0.46. While the mAP50 is lower compared to the accuracy and precision values, it provides insights into the model's ability to correctly localize and classify objects under varying thresholds.

Figure 7 depicts the detected fire regions labeled as `fire` with Confidence Score 0.81 and 0.72. Figure 8 shows the detection of the model on fire images labeled as `attacked_fire` with Confidence Scores 0.86 and 0.87. From the figures, we can derive that the model successfully, detects the fire and the attacked fire, with high confidence.

In addition, we produced figures that present the Grad-CAM heatmaps for the CNN Model detection to showcase the feature importance of pixels for some specific instances. Figure 11 depicts the detection of the model on normal fire images, while Figure 12 highlights the region of attacked fire images.

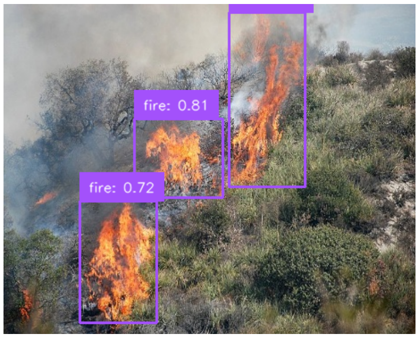

Fig. 7. Detection results on a normal fire image

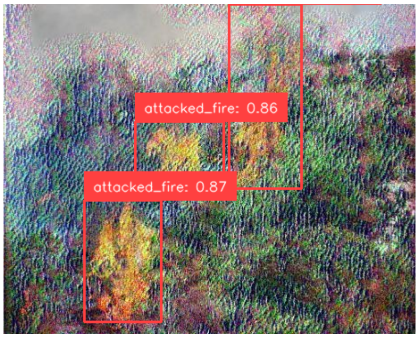

Fig. 8. Detection results on an attacked fire image

We can see how the inner workings of the classifier work when detecting fire and attacked fire instances, and we can conclude that the explanation when detecting normal fire is accurate since it highlights the area where the fire is. At the same time, the classifier learns to detect attacked fire by using pixels that are away from the pixels of the attack fire. This is a bit expected, since here we lean a classifier that tries to separate images based on two different categories. So the model learn to classify ”fire” by using fire pixels in photo, and ”attack-fire” with pixels that are away of that fire pixels.

Based on this assumption, we can say that Grad-CAM successfully explains the inner workings of the classifier and how it makes its detections. Since, we have a clear reasoning flow of how classifier separates the two classes.

Last, we observe that YOLOv8 detects and explains the attack fire pixels, while the Grad-CAM explains the attacked fire detection differently, by giving importance to pixels that are not fire pixels. This shows that both AI Models (i.e., YOLOv8 and CNN-based classifier), have very close performance but their inner workings are different.

TABLE VI
BASELINES YOLOV8 ON COCO DATASET & RESNET50 ON CIFAR-100

	Accuracy	mAP50
ResNet50	0.67	-
YOLOv8 small	-	0.446
YOLOv8 large	-	0.478

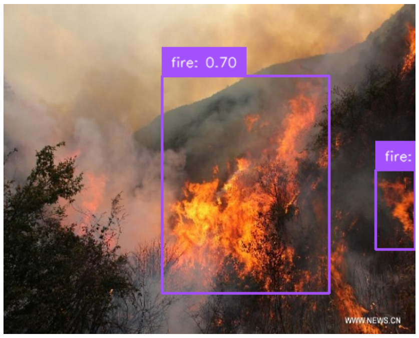

Fig. 9. Detection results on a normal fire image

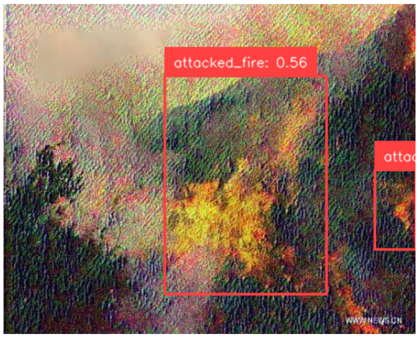

Fig. 10. Detection results on an attacked fire image

TABLE VII
RESULTS

	Accuracy	Precision	mAP50
CNN-Res50-Classifier	1.000	1.000	-
YOLOv8 small	0.925	1.000	0.460
YOLOv8 large	0.950	1.000	0.470

VII. CONCLUSIONS

In summary, combating conditions of high physical risk in cities, forests, and critical infrastructures requires a multifaceted approach. The results demonstrate that the adversarial explanations of YOLOv8 large and the CNN Model excel in accuracy and precision. The proposed approach for adversarial explanations with a Confidence Score can be a trustworthy choice for applications requiring high reliability in object detection and image classification. However, there is room for improvement in terms of mAP50, suggesting potential areas for further enhancement of the model's localization and classification capabilities.

In the future, we plan to extend the proposed Adversarial Explanations Architecture toward simulating how XAI outputs would change under different input conditions and attacks of higher complexity; and how more advanced localization features contribute to deriving more informed decisions.

VIII. ACKNOWLEDGEMENTS

This work has received funding from the Research and Innovation Foundation under Restart Research 2016-2020 Programme and the DUAL USE/0922/0024 agreement of

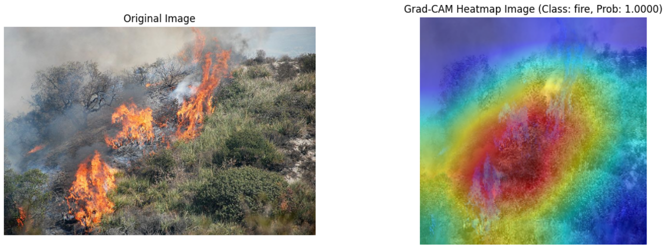

Fig. 11. Original image on the left. On the right, the image consists of the heat map of the Grad-CAM visualizing the feature (pixels) importance. Furthermore, the second image has the detection class with the Confidence Score.

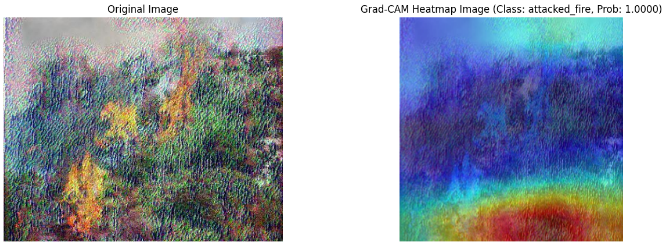

Fig. 12. Original image(attacked) on the left. On the right, the image consists of the heat map of the Grad-CAM visualizing the feature (pixels) importance. Furthermore, the second image has the detection class with the Confidence Score.

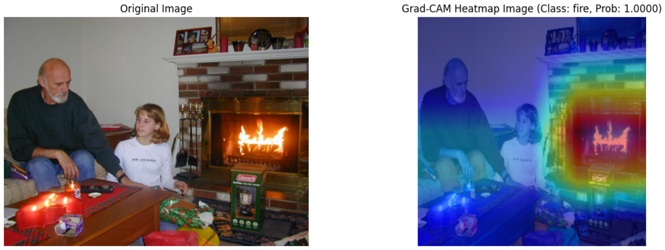

Fig. 13. Original image on the left. On the right, the image consists of the heat map of the Grad-CAM visualizing the feature (pixels) importance. Furthermore, the second image has the detection class with the Confidence Score.

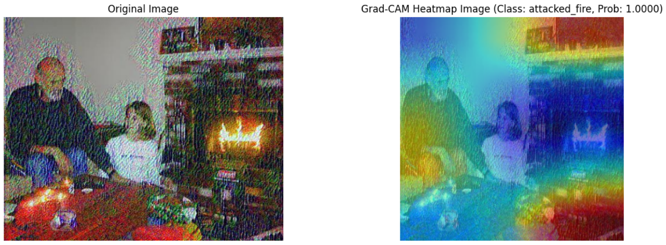

Fig. 14. Original image(attacked) on the left. On the right, the image consists of the heat map of the Grad-CAM visualizing the feature (pixels) importance. Furthermore, the second image has the detection class with the Confidence Score.

CYGNUS project, and the European Union's Horizon Europe TALON project with GA No 101070181.

References

[1] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *International conference on machine learning*. PMLR, 2019, pp. 2484–2493.

[2] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Gan dissection: Visualizing and understanding generative adversarial networks," *arXiv preprint arXiv:1811.10597*, 2018.

[3] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3681–3688.

[4] D. Stutz, M. Hein, and B. Schiele, "Disentangling adversarial robustness and generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6976–6987.

[5] J. Gao, R. Mudumbai, X. Wu, J. Yi, C. Xu, H. Xie, and W. Xu, "Towards unlocking the mystery of adversarial fragility of neural networks," *arXiv preprint arXiv:2406.16200*, 2024.

[6] A. Benchama and K. Zebbara, "Novel approach to intrusion detection: Introducing gan-mscnn-bilstm with lime predictions," *arXiv preprint arXiv:2406.05443*, 2024.

[7] Q. Card, K. Aryal, and M. Gupta, "Explainability-informed targeted malware misclassification," *arXiv preprint arXiv:2405.04010*, 2024.

[8] X. Luo, Y. Jiang, and X. Xiao, "Feature inference attack on shapley values," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 2233–2247.

[9] J. Li, M. Pang, Y. Dong, J. Jia, and B. Wang, "Graph neural network explanations are fragile," *arXiv preprint arXiv:2406.03193*, 2024.

[10] W. Woods, A. Grushin, S. Khan, and A. Velasquez, "Combining ai control systems and human decision support via robustness and criticality," in *Disruptive Technologies in Information Sciences VIII*, vol. 13058. SPIE, 2024, pp. 172–190.

[11] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[12] I. Vascotto, A. Rodriguez, A. Bonaita, and L. Bortolussi, "Can you trust your explanations? a robustness test for feature attribution methods," *arXiv preprint arXiv:2406.14349*, 2024.

[13] P. Chowdhury, M. Prabhushankar, G. AlRegib, and M. Deriche, "Are objective explanatory evaluation metrics trustworthy? an adversarial analysis," *arXiv preprint arXiv:2406.07820*, 2024.

[14] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 839–847.

[15] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," May 2024, arXiv:2305.09972 [cs]. [Online]. Available: http://arxiv.org/abs/2305.09972

[16] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Feb. 2020, arXiv:1610.02391 [cs]. [Online]. Available: http://arxiv.org/abs/1610.02391

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, arXiv:1512.03385 [cs]. [Online]. Available: http://arxiv.org/abs/1512.03385