2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 1-5, 2023. Detroit, USA

Towards a Robust Adversarial Patch Attack Against Unmanned Aerial Vehicles Object Detection

Samridha Shrestha[1], Saurabh Pathak[1] and Eduardo K. Viegas[1]

***Abstract*— Object detection techniques for autonomous Unmanned Aerial Vehicles (UAV) are built upon Deep Neural Networks (DNN), which are known to be vulnerable to adversarial patch perturbation attacks that lead to object detection evasion. Yet, current adversarial patch generation schemes are not designed for UAV imagery settings. This paper proposes a new robust adversarial patch generation attack against object detection with UAVs. We build adversarial patches considering UAV-specific settings such as the UAV camera perspective, viewing angle, distance, and brightness changes. As a result, built patches can also degrade the accuracy of object detector models implemented with different initializations and architectures. Experiments conducted on the VisDrone dataset have shown the proposal's feasibility, achieving an attack success rate of up to 80% in a white-box setting. In addition, we also transfer the patch against DNN models with different initializations and different architectures, reaching attack success rates of up to 75% and 78%, respectively, in a gray-box setting.**

GitHub: https://github.com/SamSamhuns/yolov5_adversarial

I. Introduction

Over the past decade, the market for Unmanned Aerial Vehicle (UAV) has significantly increased. According to a recent 2022 report, its market value is expected to grow in North America alone to 6.7 billion dollars by the end of 2026 [1]. UAVs have a wide range of applications, including aerial reconnaissance, search and rescue, intruder detection, and surveillance. To this extent, the manual operator control of UAVs in these complex environments may limit their application and decrease their mission efficiency. In light of this, several works have proposed new promising techniques towards the provision of autonomous UAVs, ranging from path planning [2], obstacle avoidance [3], fault detection [4], and even geolocation [5].

In such a case, object detection plays a key role in enabling autonomous UAVs to fulfill their tasks [6]. To achieve such a goal, the UAV camera feed is continuously evaluated by a Deep Neural Network (DNN) model, which reports the identified objects for the system decision-making process, such as autonomously following an identified target or reporting it back to the operator. In general, proposed schemes focus on providing the most accurate DNN-based object detector while demanding minimal processing requirements [7]. Conversely, despite their decisive impact on paving the way towards the provision of autonomous UAVs, the literature often neglects the deployed object detector's reliability and resiliency to adversaries.

[1]The authors are with Secure Systems Research Center (SSRC) at Technology Innovation Institute (TII), United Arab Emirates, Abu Dhabi {samridha, saurabh, eduardo}@ssrc.tii.ae

(a) UAV object detection without adversarial patches

(b) UAV object detection with adversarial patches

Fig. 1: Impact of adversarial patches on the performance of UAV object detection for *Cars* from the VisDrone dataset The adversarial patch was generated by training against YoloV5 Small and used to transfer attack YoloV5 Large.

In recent years, DNNs have been found to be vulnerable to adversarial attacks [8]. In practice, adversaries can significantly bias the DNN model towards detection evasion with minimal input perturbation efforts. A common attack practice relies on the application of adversarial patches, in which the adversary distorts the image pixels within a bounded-size region [9]. The adversarial patch generation process is usually optimized toward decreasing the target object detector accuracy while also accounting for better patch printability and variation aspects. The built adversarial patch can then be printed and used to evade the DNN-based object detector in the physical domain. Current adversarial patch generation techniques have demonstrated the vulnerability of general-purpose DNN-based object detectors, successfully hiding detection of people [9], traffic sign [10], or even cars [11].

In contrast, the reliability impact of the adversarial patch attack on UAV-related object detection is still in its infancy [12]. Surprisingly, current adversarial patch attacks on aerial images overwhelmingly use satellite-related imagery, which does not account for the challenges of UAV applications. UAV object detection must account for a higher range of camera perspective, angle, distance, and brightness changes, significantly increasing the adversarial patch generation efforts [11]. Correspondingly, there is still a lack of understanding on how the adversarial patches may affect the reliability of UAV object detectors. The current majority of adversarial patch threat models usually operate in a white-box setting where the adversary requires full access to the target DNN model, including its architecture, weights, and training data, with only a few reports of successful transferable attacks in the literature [13]. Consequently, most adversarial patches are only effective against a single DNN

2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-9190-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IROS55552.2023.10342460

model, usually failing at transferring their ability to affect the reliability of other object detectors.

Contributions. This paper proposes a new robust adversarial patch-generation attack against UAV object detection, implemented in two phases. First, we build adversarial patches accounting for UAV-related characteristics, including patch printability, brightness, and perspective changes. The generated patches are built in a white box setting with full access to a previously known DNN-based UAV object detector. Second, based on our robust patch generation process, we transfer the built patch to another DNN model and architecture, which is unknown during the patch generation process. As a result, our adversarial patch generation scheme can operate in a gray-box setting, wherein the adversary only requires access to a single DNN model, transferring the built patch to other object detectors.

In summary, our paper's main contributions are:

- We propose a new robust adversarial patch generation procedure against UAV object detectors. Our proposed scheme can build adversarial patches in a white-box setting with up to 80% of attack success rate.
- We experimentally evaluate the reliability of object detection on UAV domain. Our experiments have shown that adversaries can transfer their adversarial patches against different DNN models and architectures, reaching attack success rates of up to 75% and 78%, respectively.

Roadmap. The remainder of this paper is organized as follows. Section II further describes the object detection reliability on UAVs. Section III presents related works on adversarial patch generation. Section IV elaborates on our threat model, and Section V describes our proposed patch generation model. Section VI evaluates our proposed scheme, and Section VII concludes our work.

II. Preliminaries

A. Object Detection on Unmanned Aerial Vehicle

Object detection on UAVs is a widely explored topic in the literature [14], wherein related works usually conduct such a task through the implementation of four sequential modules, namely *Image Acquisition*, *Image Preprocessing*, *Object Detection*, and *Report*. First, the *Image Acquisition* module continuously collects the UAV camera images. The UAV image feed is usually collected as available on the Robotic Operating System (ROS). The image frame is then evaluated by a *Image Preprocessing* module whose goal is to preprocess the image before object detection occurs, such as conducting the image's decoding, normalization, and resizing. The built image is then evaluated by a *Object Detection* module, which identifies the image objects by applying a DNN-based object detector model. The detected objects are identified through a bounding box, which represents the location of the given object on the evaluated image. Finally, the *Report* module adequately reports the set of identified objects for subsequential decision-making on the UAV.

B. Patch Generation

Adversarial perturbations on the inputs of a DNN object detection model can significantly affect their reliability [8]. In general, currently proposed schemes conduct such a task by building adversarial patches as they can be printed and used in the physical domain. To this extent, the adversary's goal is to alter the pixels within a bounded-size region to achieve object evasion (misclassification or below detection confidence threshold) against the target model.

Given an object detector $f(x) : x \rightarrow y$ that outputs the identified object y on given an input image x. The adversary's goal is to find a patch P such that $f(x+P) \neq y$. The patch P usually follows a square-sized setting where $P \in \mathbb{R}^{s \times s \times 3}$ and s accounts for the patch size, e.g., within 30% of the target object bounding box. The adversarial patch is applied on the target object bounding box according to the adversary budget ϵ, which measures how well the adversary is able to distort the image pixels. As a result, the adversarial patch-building process is usually solved according to the following equation:

$$P(x, l) = \underset{P \in \{P' : ||P'||_\infty \leq \epsilon\}}{\arg\max} \mathcal{L}(h(A(x, l, P)); y) \quad (1)$$

where h denotes the object detector, $A(x, l, P)$ a function that applies the patch P on location l on input image x, y the set of image objects and the bounding boxes, ϵ the attack budget and $\mathcal{L}$ the object detector loss function. Consequently, the adversarial patch optimization goal is finding a patch that maximizes the object detector loss when applied to the object-bounding boxes.

III. Related Works

The vulnerability of DNN to adversarial input perturbations is a known and widely explored topic in the literature [15]. To this extent, adversarial patch attacks raise significant concern for the research community as they are also physically realizable [16]. S. Chen *et al.* [17] proposed one of the prominent approaches to generating adversarial patches against object detectors. Their work was able to conduct targeted attacks of stop signs against a state-of-the-art object detector in the physical domain. Similarly, S. Thys *et al.* [9] presented an adversarial patch attack to evade person detection in the physical domain. The authors conduct several transformations on the patch, such as improving the printability, and decreasing the pixel variation to create more "*printable*" patches. In general, achieving a robust patch that can be used in the physical domain is achieved using better transformations. As an example, S. Komkov *et al.* [18] generated a hat-style adversarial patch by applying rotation and bending transformations on the built patch.

In their vast majority, current approaches for adversarial patch generation assume a white-box setting. The adversary has full access to the target model, including its used parameters, weights, and training dataset [19]. The main challenge is that patch generation approaches often require the computation of the target model loss (see Eq. 1), making

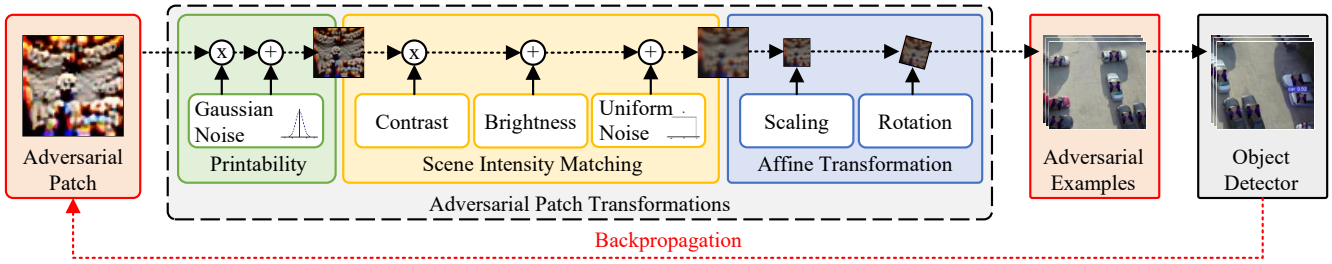


Fig. 2: Proposed robust adversarial patch generation scheme against UAV object detectors. Adversarial patch transformations address printability, scene intensity matching, and affine transformations for a more robust adversarial patch building.

the obtained patch specific to its targeted model [20]. As a consequence, several works have been proposed for the building of transferable adversarial examples. Z. Xiao *et al.* [21] aimed the evasion of face recognition by making the generated patches resemble facial expressions, improving their transferability. C. Xie *et al.* [22] improved transferability by applying several image input transformations before the patch generation.

Surprisingly, despite the impressive advancements in adversarial patch generation for general-purpose object detection, research on the UAV domain is still in its infancy [12]. Andrew Du *et al.* [11] proposed an adversarial patch generation for cars on aerial imagery. The authors conduct a sequence of transformations to address the aerial-related image aspects. Although the authors show the feasibility of adversarial patches for cars, they use satellite-related imagery, unable to depict the characteristics of visual data from a UAVs. J. Lian *et al.* [12] proposed an adversarial patch generation against aerial images of planes. Similarly, their evaluation does not consider UAV-related imagery.

IV. Threat Model

Our work considers the following threat model on adversarial patches.

Adversary's goal. The adversary's goal is to evade a set of objects from being identified as such by the UAV object detector. To achieve such a goal, the adversary must decrease the target object detection confidence below the object detector threshold. For instance, by decreasing the target object confidence to a value below 0.4.

Adversary's capabilities. The adversary has complete access to the target object detector, including its architecture, weights, and used training dataset. The adversary uses adversarial patches to evade the detection of target objects. The adversarial patch must be physically-realizable and account for the UAV characteristics. The adversarial patch must be overlaid on the target object detection location and not occlude the target object completely, for instance, by only representing 30% of the bounding box size.

V. A Robust Adversarial Patch Generation Scheme Against UAV Object Detectors

Our work goal is building adversarial patches to evade UAV object detectors in a gray-box setting. We consider a gray-box scenario wherein the attacker has white-box access to a given object detection model and aims to transfer the generated adversarial patches to another model in which the adversary has no control. More specifically, the built adversarial patches must be robust enough to be transferable to object detectors not used during the training phase. In light of this, our work formulates robust adversarial patch-building in two phases, based on an improved set of transformations and a more generalizable loss function. Figure 2 shows our proposed adversarial patch-building scheme.

First, our adversarial patch transformation scheme aims to improve patch printability and patch scene intensity matching. In contrast to related works, our patch printability loss is not designed considering a single camera setting, which usually requires finetuning for the targeted device. Conversely, we improve the patch printability by modeling the printed colors as a multivariate linear gaussian mixture of additive and multiplicative noises to the RGB image. Notwithstanding, we perform contrast, brightness, and noise adjustments on the generated patch to match it with the scene intensity. As a result, the built patch is more generalizable concerning the used adversarial patch printer and better suited for different scenes.

Second, the adversarial patch is optimized for lower object detector accuracy, smoother pixels, and lower brightness. Our main insight is building generalizable adversarial patches concerning printability and scene intensity matching to pave the way toward their transferability between object detectors.

The following subsections further describe our proposed patch generation mechanism, including its transformations and implementation aspects.

A. Adversarial Patch Transformations

Our work makes use of adversarial patch transformations to improve its robustness. The goal is to preprocess the adversarial patch to replicate the physical environment conditions before it is used against the object detector. To achieve such a goal, our proposal conducts three sets of adversarial patch transformations as shown in Fig. 2, namely *Printability*, *Scene Intensity Matching*, and *Affine Transformations*.

Printability. A printed image will not have the same coloration as its digital counterpart. The degree of variation occurs due to several factors, ranging from the type and quality of used paper, the printing device, and even the used ink. Related works address this challenge through a Non-Printability Score (NPS) added to the adversarial patch loss

computation [23]. As a result, the patch is finetuned to a single printing process in a controlled process for a specific target.

We address such a challenge using a probabilistic approach to increase the adversarial patch robustness. We model the printed colors as a multivariate linear gaussian mixture of additive and multiplicative noises to the RGB image. Let P be the square-sized in a 3-channel color adversarial patch such that $P \in [0,1]^{s \times s \times 3}$ where s denotes the patch size. We first apply the multivariate linear gaussian mixture to the adversarial patch x^* based on the following equation:

$$P = G_m \times x^* + G_a \tag{2}$$

where G_m and G_a denote the multiplicative and additive multivariate linear gaussian mixture weights, such that $G_m \in \mathbb{R}^3$ and $G_a \in \mathbb{R}^3$. The multiplicative distribution G_m reduces the intensity and contrast of the patch, whereas the additive distribution G_a changes the color distribution. As a result, the built patch is adjusted to an easier-to-be-printed format.

Scene Intensity Matching. The quality of UAV-related imagery is subject to several variations according to the scenario wherein it is collected. As a result, the used adversarial patch must adequately reflect the scene quality that it will be used. To address such a challenge, we conduct a scene intensity matching that adjusts the adversarial patch contrast, brightness, and noise. Based on the built printability adversarial patch (Eq. 2), we conduct the scene intensity matching using the following equation:

$$P = ((P \times scene_c) + scene_b) + scene_n \tag{3}$$

where $scene_c$ denotes the level of scene contrast, $scene_b$ the scene brightness, and $scene_n$ the added uniform noise. The scene intensity matching parameters can be finetuned or randomly varied to improve the built adversarial patch robustness. Henceforth, it is possible to ensure that the built adversarial patch can bias the target model in a wider range of scene characteristics.

Affine Transformations. Finally, the built adversarial patch must be adjusted for the input image object dimensions. We adjust the built patch through a scaling and rotation procedure. On the one hand, the scaling aims at adjusting the built adversarial patch to fit the target object bounding box based on a given size s. For instance, by resizing the adversarial patch to 30% of the target object bounding box. On the other hand, the rotation procedure attempts to reproduce the UAV camera variations on perspective and angle aspects.

B. Adversarial Patch Loss

Our patch optimization goal is to improve the attack success rate of the built adversarial patch while ensuring its printability and robustness. To achieve such a goal, our patch optimization is implemented through the computation of three losses, as follows:

- ***Total Patch Variation*** ($\mathcal{L}_{tv}$). The total variation loss aims to ensure that the patch pixel colors are smoother and with a better transition between them. Therefore, we compute it according to the following equation:

$$\mathcal{L}_{tv} = \frac{\sqrt{\sum_i^s \sum_j^s (P_{i,j} - P_{i+1,j})^2 + (P_{i,j} - P_{i,j+1})^2}}{N} \tag{4}$$

  where N denotes the number of pixels on the given adversarial patch P. The $\mathcal{L}_{tv}$ value is lower for similar neighbor pixels but higher for divergent ones.
- ***Total Patch Saliency*** ($\mathcal{L}_{sal}$). We use sRGB space color saliency loss that favors patches towards less vibrant or saturated colors through a quantified color metric derived from a psycho-physical color scaling user study [24]. This metric is more informative for colorfulness than saturation, which would overemphasize the patch's dark areas, helping it become less conspicuous to humans or automatic patch detection systems. We compute the total patch saliency according to the following equation:

$$\begin{gathered} rg = R - G \\ yb = 0.5 * (R + G) - B \\ \mathcal{L}_{sal} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} + 0.3 * \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \end{gathered} \tag{5}$$

  where *rg* and *yb* are variables derived from the color channel *R*, *G*, and *B* values of the patch. μ and σ represent the mean and standard deviation of the supplementary variables respectively.
- ***Total Patch Objectiveness.*** ($\mathcal{L}_{conf\_score}$). We compute the effectiveness of the adversarial patch attack against the target objective detector. To this extent, we assess the loss of the model objectiveness and classification score as follows:

$$\mathcal{L}_{conf\_score} = \frac{\sum_i^N conf(h, x_i^*, y) \times obj(h, x_i^*, y)}{N} \tag{6}$$

  where $conf$ and obj measure the object detector class confidence score and objectiveness score for the object class y for a given input adversarial image x_i^*.

Finally, we formulate our adversarial patch optimization process based on the following equation:

$$L_{patch} = \alpha L_{tv} + \beta L_{sal} + \gamma L_{conf\_score} \tag{7}$$

where α, β, and γ are hyper-parameters that denote the weights for each of our adversarial patch loss terms.

VI. Evaluation

The proposal evaluation aims at answering the following Research Question (RQ):

- (**RQ1**) *How does our adversarial patches affects the reliability of UAVs object detectors?*
- (**RQ2**) *Does our proposed robust adversarial patch building enables the attack transferability?*

The following subsections further describe the proposed model-building procedure and its evaluation.

TABLE I: Mean Average Precision (mAP) at an intersection over union threshold (IoU) of 0.5 for Yolov5 on VisDrone-2019 dataset test subset. Adversarial patches are built targeting *All* classes in a white-box setting.

DNN	Init	Patch	Classes				
			Car	Truck	Bus	People	All
YoloV5 Small	COCO	Patchless	0.87	0.46	0.64	0.68	0.66
		Random	0.70	0.14	0.25	0.53	0.40
		Adv.	0.27	0.03	0.09	0.03	0.11
	-	Patchless	0.86	0.43	0.63	0.66	0.64
		Random	0.71	0.13	0.23	0.52	0.40
		Adv.	0.30	0.04	0.11	0.10	0.14
YoloV5 Large	COCO	Patchless	0.88	0.56	0.66	0.73	0.71
		Random	0.75	0.15	0.30	0.58	0.45
		Adv.	0.41	0.04	0.13	0.06	0.16
	-	Patchless	0.88	0.51	0.67	0.71	0.69
		Random	0.73	0.14	0.28	0.58	0.43
		Adv.	0.30	0.03	0.13	0.05	0.13

TABLE II: Proposed adversarial patch building parameter variation throughout patch training phase.

Parameter Set		Parameter	Value
Patch Transformations	Printability	G_m	$N(\mu = 0.5, \sigma = 0.1)$
		G_a	$N(\mu = 0.0, \sigma = 0.001)$
	Scene Matching	$scene_c$	$[0.8, 1.2]$
		$scene_b$	$[-0.1, 0.1]$
		$scene_n$	$[-0.1, 0.1]$
	Affine	$Rotation$	$[-20°, 20°]$
		$Scaling$	30%
Patch Loss	$\mathcal{L}_{tv}$	α	2.5
	$\mathcal{L}_{sal}$	β	1.2
	$\mathcal{L}_{conf\_score}$	γ	1

A. Model Building

We consider a use-case of UAV-based surveillance to evaluate the effectiveness of our proposed adversarial patch-building scheme. To this extent, the UAV conducts the object detection on the camera feed for the identification of *Car*, *Bus*, *Truck*, and *People* objects. Our adversarial patches aim to hide the four mentioned classes from being detected as such by the UAV object detector (see Section IV). In our experiments, we consider the VisDrone-2019 [25] dataset, which has over 10 thousand static images in various resolutions and scenes acquired using a UAV platform and split across separate train, validation, and test datasets. To ensure the availability of a reasonable patching area on all the objects, we preprocess the data before training by removing objects that occupy less than 0.05% of the image area. We normalize the image values between 0 and 1 for all our experiments while using an image size of 640x640. We pad the resized images with gray pixels to keep the aspect ratio unchanged.

According to the available computational resources, we conduct the experiments based on two distinct UAV object detection platforms. On the one hand, one UAV is a resource-constrained device that uses YoloV5 Small object detector implemented with 7.2M parameters[26]. On the other hand, the other UAV conducts object detection without considering the processing constraints, hence, is implemented using the YoloV5 Large object detector model with 46.5M parameters[26]. The models can be initialized with the pre-trained weights from the well-known COCO benchmark [27] or from scratch. The models are trained using SGD optimizer with a momentum of 0.937 and a weight decay of $4\mathrm{e}{-4}$. The learning rate is linearly increased from 0.001 to 0.1 in the first three training epochs and then linearly reduced after every epoch with a learning rate scheduler that stops training if there is no improvement in the validation accuracy over 100 epochs. The object detection model was trained for 300 epochs with a batch size of 16 and implemented using PyTorch API, v.1.13.1.

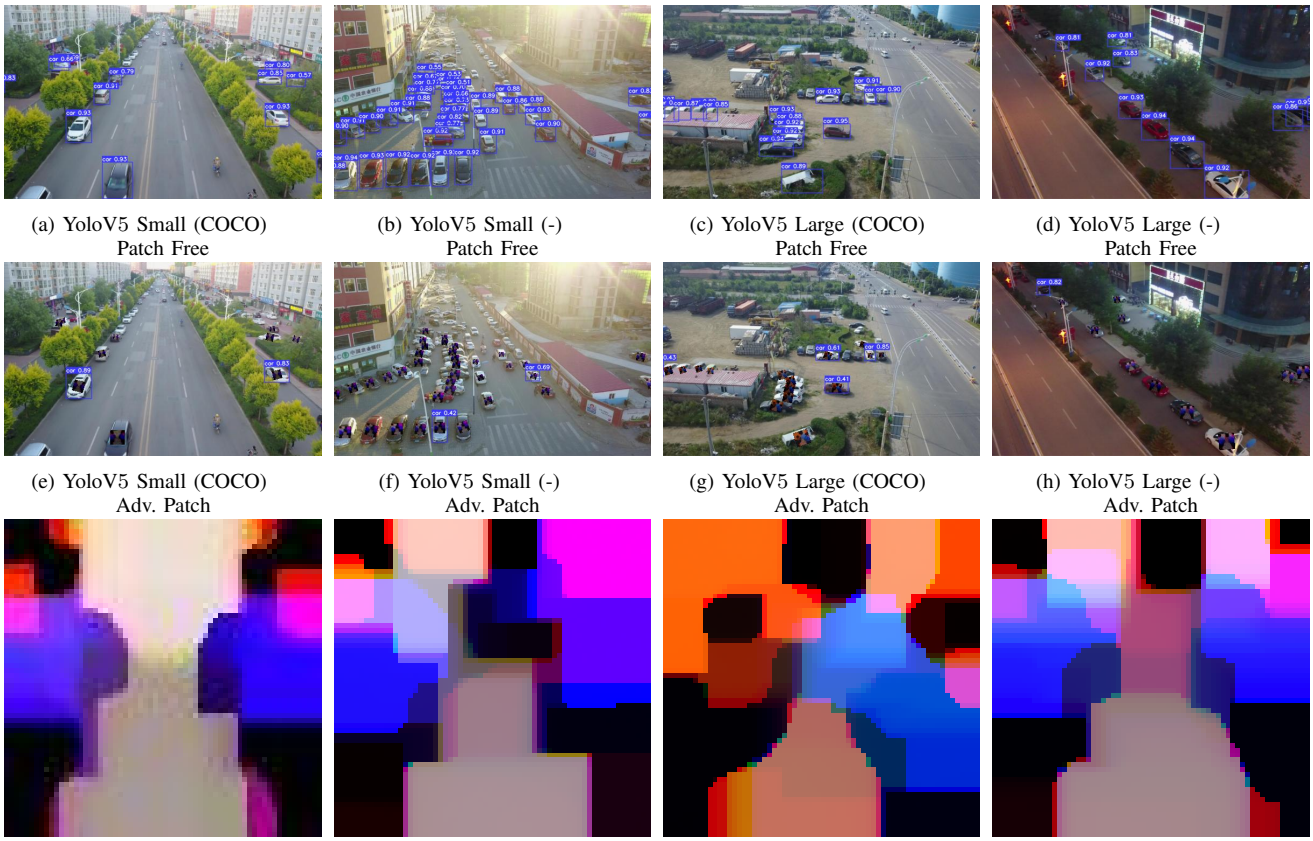
We first investigate the performance of the selected object detectors on the VisDrone dataset according to the selected object classes (*Car*, *Bus*, *Truck*, and *People*). Table I shows the object detection accuracy on the VisDrone dataset for YoloV5 Small and Large counterparts when no adversarial patches are considered (*Patchless*). We compute the Mean Average Precision (mAP) considering an Intersection over Union (IoU) threshold of 0.5. The selected object detectors generally reached an mAP higher than 0.64 when considering *All* classes (Table I). Notwithstanding, larger models can provide better detection accuracies, with an average improvement of 0.05 on their mAP compared to their lightweight counterparts. Finally, the mAP accuracy is usually related to the object occurrence on the training dataset, as noted by a higher accuracy on *Car* and *People* classes.

B. Adversarial Patches

Our second experiment aims to answer RQ1 and evaluates how our proposed adversarial patch-building scheme impacts the previously evaluated object detectors' accuracy. To this extent, we implement our proposed model (see Section V) to optimize our loss function (Eq. 7) using the Adam optimizer with a learning rate of 0.04, betas of $(0.9, 0.999)$, an ϵ of $1\mathrm{e}{-8}$ for 500 epochs. We finetune the adversarial patch transformations and loss parameters to improve their robustness throughout the experiments. We vary the printability, scene matching, and affine patch transformation parameters to achieve such a goal while optimizing the adversarial patch. Table II shows the parameters used throughout the adversarial patch experiments. The parameters were empirically set.

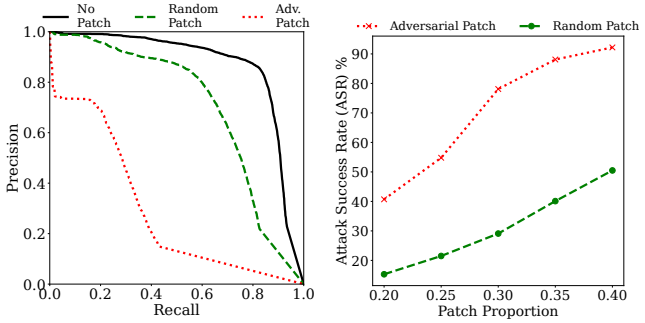
Our first proposal evaluation aim at building an adversarial patch in a white-box setting that targets *All* classes (*Car*, *Bus*, *Truck*, and *People*). The adversary has complete access to the object detection model and training dataset. We also compare our proposed model against a *Random* patch, which is filled with random noise. The goal is to measure if the accuracy drop is caused due to the added adversarial patch or due to the object occlusion. Figure 3 shows the image examples for the built adversarial patches in a white-box setting.

Table I shows the object detection accuracy with adversarial and randomly generated patches where different patches are generated based on the target class, the model architecture, and model initializations. It can be observed that adversarial patches significantly affect the reliability of the selected object detectors. The presence of adversarial patches incurred in an average mAP decrease of 0.54, representing a degradation of almost 80% for all classes across all models.



(a) YoloV5 Small (COCO) Patch Free
(b) YoloV5 Small (-) Patch Free
(c) YoloV5 Large (COCO) Patch Free
(d) YoloV5 Large (-) Patch Free
(e) YoloV5 Small (COCO) Adv. Patch
(f) YoloV5 Small (-) Adv. Patch
(g) YoloV5 Large (COCO) Adv. Patch
(h) YoloV5 Large (-) Adv. Patch
(i) Adv. Patch for YoloV5 Small (COCO)
(j) Adv. Patch for YoloV5 Small (-)
(k) Adv. Patch for YoloV5 Large (COCO)
(l) Adv. Patch for YoloV5 Large (-)

Fig. 3: YoloV5 performance samples on several VisDrone scene conditions with and without adversarial patches for the Car class. The evaluated object detection model can be trained from scratch (-) or based on the COCO dataset.



(a) Accuracy - Patch (Scaling 30%)
(b) Patch Scaling vs. ASR

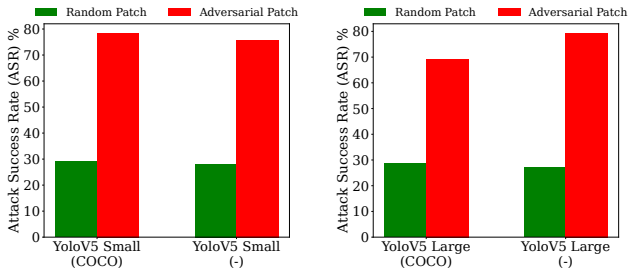
Fig. 4: Accuracy behavior of YoloV5 Small (COCO) on *VisDrone* test dataset for the *Car* class when subject to our proposed adversarial patch building technique.

The built adversarial patches can decrease the mAP by up to 0.67 in the worst case (Table I, YoloV5 Large (COCO), *People*). Additionally, it presented a similar accuracy impact regardless of the target object detector architecture and initialization parameters. For instance, it degrades the mAP by 0.53 and 0.55 for the YoloV5 Small vs. its Large counterpart, respectively. Compared to the randomly generated patches, which reduce the target object detector mAP by 37% on average, our proposed adversarial patches reduce the mAP by 80% on average, showing a 116% increase in mAP reduction. In essence, the results show that the accuracy decrease is not necessarily caused by the object occlusion but rather by the effectiveness of the added adversarial patch.

We further investigate the accuracy impact on the evaluated object detectors. Figure 4a shows the accuracy curve of one of the object detectors (YoloV5 Small (COCO)) when subject to our adversarial patch attack. It is possible to note that it dramatically impacts the target object detection accuracy compared to its randomly generated counterpart. Figure 4b shows the tradeoff on the adversarial patch scaling (size with respect to the object bounding box) vs. Attack Success Rate (ASR). To achieve such a goal we vary the *Scaling* parameter during the adversarial patch-building task (Table II). We measure the ASR according to the ratio of the previously identified objects that are not identified due to the added adversarial patch. An increase in the adversarial patch scaling significantly improves the adversary ASR. For instance, the adversary can almost double the ASR impact from 41% to 78% when increasing the adversarial patch size from 20% to 30% of the bounding box area.

C. Adversarial Patch Transferability

To answer RQ2, we further investigate the robustness of the built adversarial patches towards the transferability



(a) Initialization Transferability (b) Architecture Transferability

Fig. 5: Transferability evaluation of the built adversarial patches on *VisDrone* test dataset for the *Car* class. The adversarial patch is built in a white-box setting targeting a YoloV5 Small architecture with a COCO-pretrained weight initialization approach. The resulting adversarial patch is then used against other object detector initialization and architectures.

between different object detector model initialization and different architectures. We use the adversarial patch built against the YoloV5 Small architecture for the *Car* class against a different object detector model initialization and architecture to achieve such a goal. We choose the *Car* class since it represents nearly 60% of data samples from our filtered VisDrone dataset while the other classes have very small visual sizes, i.e. people, or have fewer samples in the dataset, i.e. truck, bus.

Figure 5a shows the ASR of our built adversarial patch when used against the same object detector architecture but with a different initialization procedure. The sustained ASR across the YoloV5 Small models shows the transferability of our adversarial patch against an object detector with different initializations. Transferring the adversarial patch to another model caused a non-significant reduction in the ASR of only 3.1% from 78.1% to 75.7%. As a result, the adversary only needs prior knowledge about the object detector architecture implemented on the UAV. This is because the attacker can transfer the adversarial patch built against the same architecture, but implemented with different initialization criteria, thus, with different weights. This enhanced robustness of our proposed adversarial patch-building approach significantly increases the threat against autonomous UAVs.

We also investigate if the built adversarial patches can be transferred to another object detector architecture of a similar family. Figure 5b shows the ASR when we apply the adversarial patch from YoloV5 Small initialized from COCO-pretrained weights against the YoloV5 Large architecture. We observe the built adversarial patches are robust against different object detector architectures. In this case, using the adversarial patch built for YoloV5 Small architecture against YoloV5 Large incurs in a ASR drop of 11.6% from 78.1% to 69.0% for the scratch initialized and a non-significant increase of 1.2% from 78.1% to 79.0% for COCO pre-trained model. These changes in ASR are within the error thresholds of our repeated experiments signifying no significant reduction or increment in patch effectiveness across different models of similar architectural families. This transferability is most likely due to the fact that even different models of similar architectural families (i.e. YoloV5 Small vs YoloV5 Large) tend to use similar detection heads and necks but with a varying number of feature extraction layers which ultimately attend to similar features in images. Consequently, our proposed adversarial patch against object detectors is robust against variations not only on the used object detector initialization scheme but also on different architectures.

VII. Conclusion

DNN-based object detection is essential for processing the huge amounts of image data generated by autonomous UAVs systems. To this extent, ensuring their safety, resiliency, and robustness against adversarial attacks is a must. This paper has proposed a new adversarial patch-building procedure against UAV object detection to improve the built patch's robustness in a more realistic setting where an attacker does not have access to the target model's internal parameters. Our experiments, using a UAV imagery dataset, have shown that our proposed technique can significantly affect the reliability of current UAV object detectors. Furthermore, the built adversarial patches can also be transferred to DNN models with different initializations and different architectures, significantly increasing the real threat posed by such adversarial patches and easing the attacker's job to evade detection from these UAV systems. Our findings have significant implications for the security of autonomous UAV-based object detection systems and provide insights into the weakness of DNN models against adversaries in UAV applications. In future works, we plan to port the generated adversarial patches across different object detector architecture families and to the physical domain.

References

[1] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the unmanned aerial vehicles (uavs): A comprehensive review," *Drones*, vol. 6, no. 6, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/6/147

[2] Q. Kuang, J. Wu, J. Pan, and B. Zhou, "Real-time uav path planning for autonomous urban scene reconstruction," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1156–1162.

[3] H. Chen and P. Lu, "Computationally efficient obstacle avoidance trajectory planner for uavs based on heuristic angular search method," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5693–5699.

[4] S. S. Katta and E. K. Viegas, "Towards a reliable and lightweight onboard fault detection in autonomous unmanned aerial vehicles," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1284–1290.

[5] M. Jun, Z. Lilian, H. Xiaofeng, Q. Hao, and H. Xiaoping, "A 2d georeferenced map aided visual-inertial system for precise uav localization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4455–4462.

[6] M. Franke, C. Reddy, D. Ristić-Durrant, J. Jayawardana, K. Michels, M. Banić, and M. Simonović, "Towards holistic autonomous obstacle detection in railways by complementing of on-board vision with uav-based object localization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7012–7019.

[7] S. Alabachi, G. Sukthankar, and R. Sukthankar, "Customizing object detectors for indoor robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8318–8324.

[8] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1369–1378.

[9] S. Thys, W. Van Ranst, and T. Goedeme, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[10] X. Wei, Y. Guo, and J. Yu, "Adversarial sticker: A stealthy attack method in the physical world," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2711–2725, 2023.

[11] A. Du, B. Chen, T.-J. Chin, Y. W. Law, M. Sasdelli, R. Rajasegaran, and D. Campbell, "Physical adversarial attacks on an aerial imagery object detector," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1796–1806.

[12] J. Lian, S. Mei, S. Zhang, and M. Ma, "Benchmarking adversarial patch against aerial detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[13] X. Wei, Y. Guo, J. Yu, and B. Zhang, "Simultaneously optimizing perturbations and positions for black-box adversarial patch attacks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–13, 2022.

[14] T. Ye, W. Qin, Y. Li, S. Wang, J. Zhang, and Z. Zhao, "Dense and small object detection in uav-vision based on a global-local feature enhanced network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[15] A. R. Ba Nabila, E. K. Viegas, A. Almahmoud, and W. T. Lunardi, "A generative adversarial network-based attack for audio-based condition monitoring systems," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 2023, pp. 275–280.

[16] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 665–681.

[17] S.-T. Chen, C. Cornelius, J. Martin, and D. H. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*. Springer, 2019, pp. 52–68.

[18] S. Komkov and A. Petiushko, "Advhat: Real-world adversarial attack on arcface face id system," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 819–826.

[19] A. Braunegg, A. Chakraborty, M. Krumdick, N. Lape, S. Leary, K. Manville, E. Merkhofer, L. Strickhart, and M. Walmer, "Apricot: A dataset of physical adversarial attacks on object detection," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 35–50.

[20] A. Liu, J. Wang, X. Liu, B. Cao, C. Zhang, and H. Yu, "Bias-based universal adversarial patch attack for automatic check-out," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 2020, pp. 395–410.

[21] Z. Xiao, X. Gao, C. Fu, Y. Dong, W. Gao, X. Zhang, J. Zhou, and J. Zhu, "Improving transferability of adversarial patches on face recognition with generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11 845–11 854.

[22] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[23] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1528–1540. [Online]. Available: https://doi.org/10.1145/2976749.2978392

[24] D. Hasler and S. Suesstrunk, "Measuring colourfulness in natural images," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5007, pp. 87–95, 06 2003.

[25] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7380–7399, 2022.

[26] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7347926

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.