1. **AP-PA算法介绍**

AP-PA（adaptive-patch-based physical attack）算法旨在在真实的物理场景下，生成一种鲁棒性的对抗性补丁，该对抗性贴片能轻松实现隐身攻击效果;此外，对抗性贴片对复杂的物理变化具有鲁棒性，可以灵活方便地使用。本章节中

**1.1基本原理**

在航空探测任务中，我们面临的挑战是如何设计一种对抗性攻击，以使探测系统忽略恶意植入的航拍图像中的目标对象。这种攻击的关键在于创建一个对抗性补丁，它不仅要在实际应用中表现出强大的攻击能力，而且要具备普适性，能够应用于目标对象之外的场景。为此，我们采取了一种独特的方法：在训练过程中固定目标航空探测器的权重和偏差，而迭代更新对抗性补丁的像素值，实际上是在“训练”这个补丁，而非整个包含大量飞机图像的数据集。

从数学上讲，具有详细补丁的对抗性示例可以表述为：



其中和分别表示 Hadamard 乘积和对抗性补丁。掩模矩阵用于限制对抗性补丁的大小、形状和位置，其中补丁位置区域的值为1。

首先，我们从一个包含随机噪点的原始的对抗性补丁开始，通常这个补丁为随机灰度图像。这个补丁经过精心设计的一系列物理和尺度自适应变换，以确保其在不同环境下的适用性。经过这些变换，补丁将被放置在一张未受污染的航拍图像的检测物体（在本实验中为飞机）上方以保证覆盖物体，从而形成一个具有潜在对抗性的图像。这一步骤是构建有效对抗性攻击的基础。

随后，我们将这个经过改造的图像输入到目标航空探测器中，以评估其对探测器性能的影响。探测器的响应——特别是其检测结果——将被用来提取一个关键的量化指标：客观性分数。这个分数不仅反映了探测器对图像中目标的识别能力，而且将成为我们评估对抗性攻击效果的重要依据之一。

我们将利用这个客观性分数来构建一个损失函数，该函数将指导我们的优化过程。该损失函数包括对抗性客观性损失这（）、不可打印性评分（）和总变异损失（）。通过反向传播算法，我们将对对抗性补丁的像素值进行调整，以期最小化损失函数的值。这个过程是迭代的，每一次迭代都会使补丁更加接近我们的目标：在不引起探测器警觉的情况下，有效地干扰其对目标的识别。最终，重复上述步骤，我们将得到一个经过充分训练的对抗性补丁。AP-PA整体架构如图3-11。



图 3-11 AP-PA工作流程

|  |
| --- |
| **算法1： AP-PA** |
| **0:输入：**  **检测器𝐷(⋅), 良性的空中图像 𝒙 以及对应的地面真实标签 𝒚, 原始的补丁 𝒑0, 对抗攻击损失函数 𝐿(⋅), 训练周期总数𝑁𝑒𝑝𝑜 以及每个周期迭代次数𝑁𝑖𝑡𝑒, 图像尺寸 𝑠, 超参数𝛼,𝛽,𝜂**  **输出：对抗性补丁P\***  **1:  Initialize 𝒑0 randomly in [0, 255], 𝒑∗=𝒑0;**  **2:  for 𝑖=0 to 𝑁𝑒𝑝𝑜 do**  **3:     for 𝑗=0 to 𝑁𝑖𝑡𝑒 do**  **4:        Patch transformation,**  **5:        𝒑∗=𝑃𝑇(𝒑∗);**  **6:        Patch applier,**  **7:        𝒙∗=𝑃𝐴(𝒑∗,𝒍𝒑∗,𝑤𝒑∗,h𝒑∗);**  **8:        Detection,**  **9:        𝒓=𝐷(𝒙∗);**  **10:        Extract objectiveness loss,**  **11:        𝐿𝑜𝑏𝑗=𝐸(𝒓);**  **12:        Total loss,**  **13:        𝐿=𝐿𝑜𝑏𝑗+𝛼⋅𝐿𝑡𝑣+𝛽⋅𝐿𝑛𝑝𝑠;**  **14:        Update patch,**  **15:        𝒑𝑖,𝑗+1∗=𝒑𝑖,𝑗∗+𝜂⋅∇𝒑𝑖,𝑗∗𝐿;**  **16:     end for**  **17:  end for**  **18:  𝒑∗=𝒑𝑁𝑒𝑝𝑜,𝑁𝑖𝑡𝑒∗;**  **19:  return 𝒑∗** |

**1.2补丁适配**

为了使AP-PA算法制作的对抗性补丁在真实场景中成功欺骗空中探测系统，我们在优化对抗性补丁的过程中适应了物理世界的动态条件。真实场景通常包含不同的条件，包括动态视角、自然噪点、明暗的变化等。我们采用几种物理变换来模拟这些动态因素。从技术上讲，我们考虑了适应物理扰动条件的转换，例如添加噪声、变化比例、随机旋转、照明偏移等。以上物理自适应操作都打包在补丁转换函数中，则对抗性示例可以写成：



我们的目标是在攻击对象的中心放置一个自适应大小的对抗性补丁。通过使用检测结果中的（）坐标来进行中心坐标的计算（公式略）。接着，我们考虑到物体的不同尺度，提出一种尺度自适应贴片方法来解决这个问题。要使对抗性补丁和目标对象的面积保持适当的比例 。

然后，我们要通过对函数建模来说明这些参数的关系，该函数旨在将对抗性补丁粘贴在具有自适应大小的相应中心位置。这样的话原始的公式问题将转化为：



**1.3对抗性客观损失**

将非最大抑制操作后检测到的对象的所有客观性得分的平均值作为对抗性客观损失，可以写成：



其中r为航空探测器的探测结果，表示从 r包含 n个探测物体的物体中提取客观性损失 ，包括每个物体的坐标（） 、检测客观分数和类分数等。对抗性补丁的目的是将飞机隐藏在航拍图像中。为了实现这一点，我们的目标是降低航空探测器预测的物体或类别分数。我们之所以在损失函数中不考虑分类分数，是因为最小化飞机的分类分数往往会增加不同等级的分数。将分类分数考虑在内并不能获得更强的攻击力。

**1.4 不可打印性损失**

由于AP-PA算法针对物理层面的颜色与图像进行攻击，我们生成的对抗性贴片应该具有可打印性以适应物理颜色，因此我们引入不可打印性损失来描述我们的对抗性补丁在物理世界中可以打印的效果，它表示了普通打印机打印的数字域和物理世界之间的对抗性贴片的距离。



**1.5 总变异损失**

为了捕捉相邻像素之间的值，以防相邻像素间值过大，我们引入总变异损失，顾名思义，该参数旨在抑制相邻像素之间的值“变异”，即变化过大。这将使我们的优化器偏好于更加平滑且颜色偏移量较小的对抗性补丁，以下是总变异损失的公式：



在上述三个分量中，总目标函数如下：



我们使用超参数 α和 β分别进行缩放 和 相加，然后 用Adam 进行优化。我们提出的AP-PA旨在最小化目标函数并优化对抗性补丁，因此我们冻结了空中检测模型中的所有权重和偏差，只更新了对抗性补丁的像素值。初始补丁 p0在优化过程开始时会提供随机值。

**2．RPAttack测试与结果分析**

1、配置环境

Python3.8

Cuda11.3

pytorch==1.11.0

Python packages：分别配置mmdetection和yolov5所需的库

训练用预训练模型：使用dota数据集进行训练的faster r-cnn模型

训练用数据集：RSOD数据集

2、将训练好的Faster-rcnn检测网络的权重以及相关参数替换为对应的文件和参数。

3、运行train\_patch\_mmdetection.py文件（包含生成攻击主函数），调整网络中相关参数（如epoch等）

其中我们将epoch设置为500，批次batch为1，在测试和训练中，所有航空探测器的并集交集 （IOU） 和客观性置信度阈值分别为 0.45 和 0.4。此外我们将超参数α与β分别设置为2.5和0.01。

文件中主要功能函数如下：

PatchTrainer:核心功能。主要训练函数，包含读取图像、生成初始补丁灰度图像、初始化模型、初始化推理器、初始化优化器和学习率调度器等，并使用wandb进行实验跟踪。

patch\_transformer：核心功能。对给定的对抗补丁应用一系列数据增强操作，包括旋转、缩放和随机位置调整。这些操作旨在增强模型的泛化能力，使其更难以被对抗样本欺骗。

输入：参数:

- adv\_patch: 输入的对抗补丁，需要应用数据增强的操作。

- lab\_batch: 标签批次，包含目标物体的位置和大小信息。

- img\_size: 目标图像的大小，用于调整补丁和标签的大小以适应目标图像。

- do\_rotate: 是否执行旋转数据增强，默认为True。

- rand\_loc: 是否在目标物体的位置上添加随机偏移，默认为True。

返回：经过数据增强后的对抗补丁，大小和目标图像一致，并应用了旋转、缩放和随机位置调整。

PatchApplier：先将非补丁位置的数adv设为1，然后在adv为0时应用补丁像素。旨在将补丁打在给定的精心选择的位置上。

NPSCalculator：用于计算NPS损失。

TotalVariation：用于计算TV损失。

InriaDataset：读取img\_dir与lab\_dir中的图像和标签并一一对应。

BaseConfig：用于配置训练集图像，训练集标签，用于攻击的预训练模型，学习率，补丁大小，训练批次等基本参数。

faster\_rcnn：初始化推理检测器实例，初始化检测模型，加载预训练模型权重，定义模型结构和训练参数等功能。

4、运行结果：

在配置好AP-PA网络之后，我们利用DOTA数据集训练出的Faster-rcnn和作为检测器，对RSOD数据集中的图像生成对抗补丁并实时应用于图像中，以攻击检测网络。我们将生成的攻击图像进行旋转、缩放、灰度变换等操作，并将得到的对抗补丁进行轻微的位置移动，分别模拟现实世界中拍摄角度、拍摄高度、光线以及对抗物体位置偏移等情况，最后对这些变换后的对抗图像进行检测，以验证AP-PA攻击方法对现实世界中由于拍摄角度、高度、光线和位置变化不同而得到的不同检测目标图像的鲁棒性，实验结果如图。

**（1）对抗图效果及鲁棒性分析**

|  |  |  |
| --- | --- | --- |
| result_0_21 | result_0_85 | |
| result_0_66 | result_0_104 | |
| result_0_109 | result_0_351 | |
| result_0_241 | | result_0_250 |
| result_0_261 | | result_0_281 |
| 图 3-1 AP-PA生成对抗样例 | | |
| adv_patch_54_400 | | |
| 图 3-2 对抗性补丁，大小为50\*50 | | |

图3-1展示了AP-PA生成唯一的对抗性补丁，该补丁在最终生成后并未对图片进行自适应优化，而仅对图片中的飞机进行自适应的覆盖，始终覆盖在飞机正中心并进行随机偏移与随机旋转。所有图片均在生成时已进行随机的对比度、亮度、旋转角度、清晰度、随机噪点等处理。部分结果如图3-1所示，可以看出该补丁大幅降低了检测的准确率，部分已完全无法检测，一小部分被识别为直升机。此外，通过对实验结果进行分析，我们总结了部分显然而然的结果，即

1. 当亮度适宜且对比度较高时，物体更容易被检测。
2. 清晰度越高，物体越容易被检测到。
3. 相机-物体距离越小，物体越容易被检测到。

例如在第10张图中，很明显可以观察到亮度的分界线，而在左侧亮度较低的环境中，所有飞机都无法被检测到；此外，靠近相机的飞机将更容易被检测到，而距离较远的飞机更容易被忽略或被视为直升机。

此外，通过观察det\_loss，可以清楚地知道到随着训练次数的增加，准确率逐渐降低至40%左右，成功达到攻击目标，我们认为这是合格的鲁棒性的攻击方式。

|  |
| --- |
| W&B Chart 2024_7_4 02_40_09 |
| 图 3-2 平均det\_loss |

相对于前两种攻击方式，AP-PA同时具有极高的生成速度与较高物理应用的可能性两种优点，通过在物理世界中使用类似喷漆或者布条等方式进行对抗性补丁的覆盖，且无需考虑方向性，即可用于伪装并躲避检测模型的检测。此外，由于AP-PA通过高随机性的改变各种物理条件进行训练，对实际物理世界中光照、对比度、相机-物体高度差等影响检测的参数具有较高的鲁棒性。

此外，AP-PA生成对抗样本的效率远高于DPAttack，但攻击效率低于DPAttack，具有不稳定的特点。我们推测是补丁大小无法覆盖全部飞机，补丁大小限制了对目标检测模型的攻击性。相较于DPAttack的针对性攻击，AP-PA攻击无法适应所有环境，特别是高清晰度、高对比度的图像。如图3-3，不同环境、不同图像的对抗性样本图像检测准确率从0-98%不等，平均准确率大约在37%左右。

|  |
| --- |
| W&B Chart 2024_7_4 03_15_53 |
| 图 3-3 对抗样本集每张图像的det\_loss |