PatchCleanser: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier

Abhinaya Sree Talluri
Clemson University

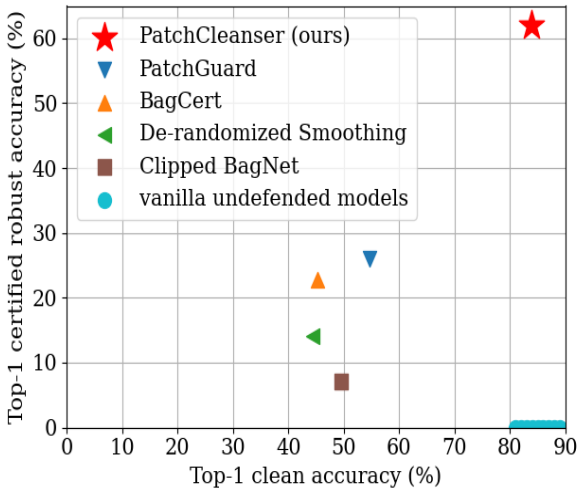
Sravanth Revath Krishna Thatavarthi
Clemson University

Abstract

The adversarial patch assault is a technique for tricking image classification models by including a tiny patch of intentionally misclassified pixels to an image. By affixing the patch to the object being classed, this kind of assault can be carried out in the real world, endangering computer vision systems. As a strong defense against hostile patches, PatchCleanser has been created as a solution to this problem. To combat the impacts of the adversarial patch, the method uses two rounds of pixel masking on the input image.PatchCleanser can efficiently collaborate with any cutting-edge image classifier and achieve high accuracy thanks to its picture-based operation. Additionally, the defense can be deemed reliable because it will consistently anticipate the correct class labels on particular images against any adaptive white-box attacker inside the specified threat model. On the ImageNet, ImageNette, and CIFAR-10 datasets, PatchCleanser has undergone extensive testing and shown comparable clean accuracy to cutting-edge classification models as well as significantly increased certified robustness over prior methods.Notably, for the ImageNette dataset, PatchCleanser obtained 99.6% top-1 clean accuracy and 96.4% top-1 certified robust accuracy against a 2%-pixel square patch found anywhere on the image.

1 Introduction

The adversarial patch attack is designed to cause test-time misclassification of image classification algorithms. An attacker who uses a patch injects maliciously created pixels into a small, constrained area (i.e., a patch) and can carry out a physical attack by printing and affixing the patch to the target object. [1] [6] Patch attacks provide a serious danger to real-world computer vision systems since they are physically feasible. There has been ongoing research on defenses against adversarial patches to protect the deployment of crucial computer vision systems. These defenses are designed to provide accurate predictions on specific images even in the presence of an adaptive whitebox attacker with a certifiable guarantee. [3] This powerful robustness trait offers a method to put an end to the arms race between defenders and attackers.The adversarial patch attack is a technique used to deceive image classification algorithms into misclassifying images. In this attack, a patch of maliciously created pixels is added to a portion of the image with the aim of causing the model to misclassify the image during testing. In real-world scenarios, a patch attacker can print and attach the patch to the object being recognized to execute the attack. As a result, patch attacks pose a significant threat to computer vision systems that are deployed in practical settings. This issue has been studied previously, and it is evident that robust defenses are necessary to protect against patch attacks. [7]



2 Problem Statement

The problem addressed in the paper "PatchCleanser: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier" is the threat of adversarial patch attacks against image classification models. Adversarial patch attacks involve injecting a patch of adversarially crafted pixels into a specific region of an image, with the goal of causing misclassi-

fication by the model during testing. The attacker can execute this attack in the physical world by printing and attaching the patch to the object being recognized. [9] Adversarial patch attacks pose a significant threat to computer vision systems deployed in real-world settings. The paper proposes a defense mechanism called PatchCleanser, which aims to neutralize the effect of the adversarial patch by performing two rounds of pixel masking on the input image. The authors demonstrate that PatchCleanser achieves high accuracy and certified robustness against adaptive white-box attackers within their threat model, making it a promising defense against adversarial patch attacks. [15]

PatchCleanser is an architecture-agnostic, certifiably robust defense against adversarial patches in image classification. It does not rely on abstention, which is a technique in which a classifier refuses to make a prediction if it is uncertain about the correct label, as it can negatively impact accuracy. [**?**] PatchCleanser works by performing two rounds of pixel masking on the input image to neutralize the effect of the adversarial patch. The first round of masking applies a circular mask around the patch to remove all pixels within a certain distance of the patch. The second round of masking removes a slightly larger region of pixels to ensure that any residual features of the patch are fully removed from the image. [1] [14] [7] PatchCleanser is designed to be compatible with any state-of-the-art image classifier, allowing for high accuracy. Additionally, it achieves certified robustness by proving that it will always predict the correct class labels on certain images against any adaptive white-box attacker within the threat model. Extensive evaluations on the ImageNet, ImageNette, and CIFAR-10 datasets have demonstrated that PatchCleanser achieves similar clean accuracy as state-of-the-art classification models and significantly improves certified robustness from prior works. [4]

3 Related Work

A defense mechanism called PatchCleanser was created to offer an objectively robust response to adversarial patches for any image classifier. In order to offset the effects of the adversarial patch and neutralize their influence, the technique uses two rounds of pixel masking on the input image. Due to the operation's image-based nature, PatchCleanser can be used with any cutting-edge image classifier and achieve excellent accuracy. The defense mechanism can also be deemed reliable because it has the ability to anticipate the proper class labels on specific photos in the face of any adaptive white-box attacker operating inside a predetermined threat model. [16]On the ImageNet, ImageNette, and CIFAR-10 datasets, PatchCleanser has been tested. The testing findings showed that PatchCleanser can provide clean accuracy that is comparable to current state-of-the-art classification models while greatly enhancing certified robustness compared to earlier methods. The results were astounding: for the 1000-class ImageNet dataset, PatchCleanser achieved 83.9% top-1 clean accuracy and 62.1% top-1 certified robust accuracy against a 2%-pixel square patch anywhere on the image.

3.1 Adversarial Patch Attacks:

Adversarial patch attacks are a type of attack in which a small, specifically designed patch is added to an image or object in order to fool machine learning models into misclassifying the object. These attacks are a form of adversarial attack, which aim to exploit weaknesses in machine learning models. The patch is designed to be visually imperceptible to humans, but it can be detected by machine learning algorithms. [13]The patch is strategically placed in an image or object so that when the image is classified, the algorithm will misclassify it with high confidence. Adversarial patch attacks can be used to trick machine learning algorithms in a variety of applications, including self-driving cars, facial recognition, and object recognition. One of the main challenges with defending against adversarial patch attacks is that the patches are often specifically designed to target a particular model, and can be difficult to detect. [17]Researchers have developed several techniques for defending against these attacks, including retraining models on adversarial examples and using input transformations to make the patch less effective. However, these defenses are not foolproof, and adversarial patch attacks remain a significant threat to machine learning models.

3.2 Adversarial Patch Defenses:

Defending against adversarial patch attacks is a challenging task, as the patch can be designed to be visually imperceptible to humans and can fool machine learning models into misclassifying an image. [18] However, researchers have developed several techniques to mitigate the impact of these attacks. Here are some common defenses against adversarial patch attacks:

1. Robust Training: One of the most effective defenses is to train the machine learning model with adversarial examples during the training phase [5]. This helps the model to learn to be robust against adversarial attacks.

2. Input Transformation: Another defense is to apply random perturbations to the input image or object before it is processed by the machine learning model [16]. This can make the adversarial patch less effective, as the perturbations can mask the patch's presence.

3. Patch Detection: Another defense is to detect the presence of adversarial patches in the input image. This can be achieved by training a separate model that is specifically designed to detect adversarial patches, or by using a pre-trained model that has been shown to be effective at detecting patches [3].

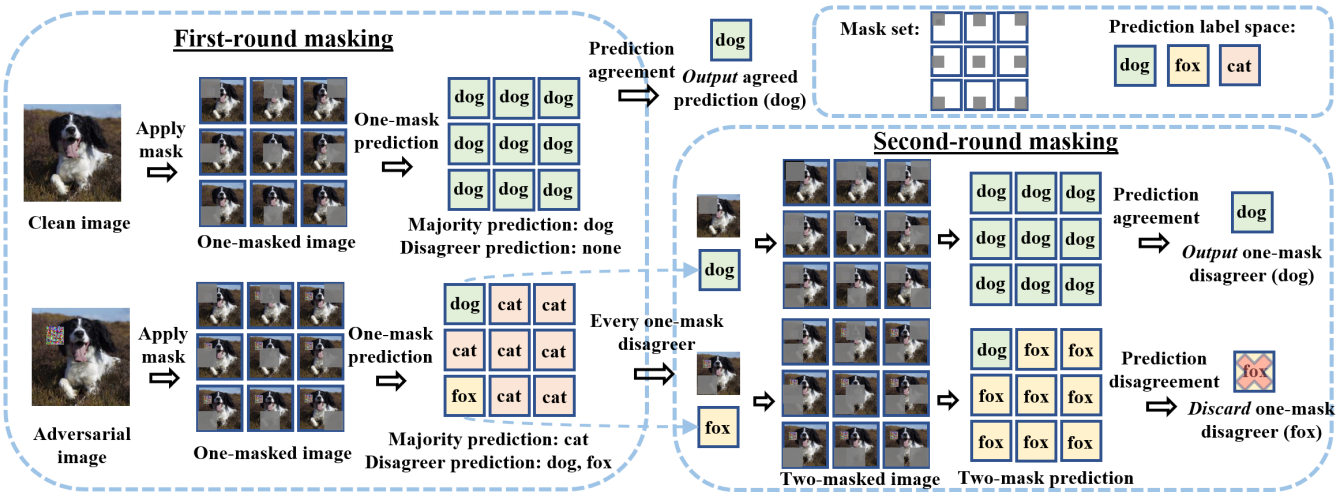


Figure 1: Overview of double-masking defense: Double-masking defense is a type of defense against adversarial patch attacks in which two rounds of masking are performed on the input image to neutralize the effect of the adversarial patch.In the first round of masking, a circular mask is applied around the patch. This mask removes all pixels within a certain distance of the patch, effectively erasing the patch and any of its features from the image. However, this masking can still leave some residual features from the patch that may be detected by the image classifier.To address this, a second round of masking is applied, in which a slightly larger region of pixels is removed from the image. [11] This ensures that any remaining features of the patch are fully removed from the image, and only the true features of the object are used for classification.

4. Model Verification: Finally, model verification techniques can be used to verify that the machine learning model is behaving correctly and has not been compromised by an adversarial patch [15].

It is important to note that while these defenses can help mitigate the impact of adversarial patch attacks, they are not foolproof. Adversarial patch attacks remain a significant threat to machine learning models, and research in this area is ongoing.

3.3 Other Adversarial Example Attacks:

Adversarial example attacks are a class of attacks that aim to deceive machine learning models by introducing small, carefully crafted perturbations to input data. These perturbations are often visually imperceptible to humans but can cause machine learning models to misclassify the input data with high confidence [4]. Here are some other types of adversarial example attacks:

1. FGSM (Fast Gradient Sign Method): FGSM is a type of adversarial example attack that generates perturbations by computing the gradient of the loss function with respect to the input data and using the sign of the gradient to perturb the input [13].

2. DeepFool: DeepFool is an iterative algorithm that generates adversarial examples by finding the minimum perturbation required to misclassify an input data point [2].

3. Carlini-Wagner (CW) Attack: The CW attack is a powerful attack that is capable of generating adversarial examples with high confidence even when the attacker has limited knowledge about the target model. The CW attack uses an optimization algorithm to find the perturbation that minimizes the distance between the adversarial example and the original input while maximizing the probability of misclassification [5].

4. Universal Adversarial Perturbation: A universal adversarial perturbation is a perturbation that can be added to any input data point to cause misclassification. These perturbations are typically generated by iteratively updating a single perturbation that is applied to multiple input data points [7].

These attacks are significant threats to machine learning models and can have serious consequences in applications such as autonomous vehicles, facial recognition, and fraud detection. As a result, researchers are actively working to develop robust defenses against these attacks.

4 Methodology

The adversarial patch attack against image classification models is something that the defense mechanism PatchCleanser is intended to stop. This method seeks to provide certifiable robustness against any adaptive white-box attacker, high accuracy, compatibility with cutting-edge image classifiers, and

efficacy in the real world.The objective of the PatchCleanser methodology is to reduce the influence of the adversarial patch via two rounds of pixel masking on the input image. The patch's location inside the image is identified in the first round by a binary mask that is made. A random mask is created in the second round and applied to the entire image [9]. This mask is created to make sure that the adversarial patch does not significantly affect the classification outcome of the model.The main benefit of PatchCleanser is that it achieves robustness that can be verified. To put it another way, it is mathematically demonstrated that it will consistently predict the proper class labels on particular photos against any adaptive white-box attacker inside the stated threat model [10]. This makes it a particularly compelling defense mechanism against adversarial patch attacks.PatchCleanser has been extensively evaluated on multiple datasets, including ImageNet, ImageNette, and CIFAR-10. The results demonstrate that the defense achieves similar clean accuracy to state-of-the-art classification models and significantly improves certified robustness over previous approaches.

PatchCleanser Design:

PatchCleanser is a defense mechanism designed to detect and remove adversarial patches from images. It consists of two components: a patch detector and a patch remover. The patch detector component uses a neural network to classify an input image as either "patched" or "unpatched". The neural network is trained on a dataset of images with and without adversarial patches. During inference, the patch detector takes an image as input and outputs a binary prediction of whether the image has been patched with an adversarial patch [8]. The patch remover component is responsible for removing the detected adversarial patch from the image. It works by first identifying the location of the patch in the image using a technique called saliency maps. Saliency maps highlight the most important regions of an image, and can be used to determine the location of the adversarial patch. Once the patch location has been identified, the patch remover applies a process called inpainting to fill in the patch region with content from the surrounding areas of the image.PatchCleanser is evaluated based on its ability to correctly detect and remove adversarial patches while preserving the accuracy of the underlying image classification task. The performance of PatchCleanser is typically measured using metrics such as the detection rate (percentage of patched images correctly identified by the patch detector) and the image classification accuracy (percentage of images correctly classified after the patch has been removed) [7]. The effectiveness of PatchCleanser may vary depending on the type of adversarial patch being used and the target model architecture, and ongoing research is exploring ways to improve the performance of this defense mechanism.

Pixel Mask Set:

Pixel Mask Set is a defense mechanism that aims to prevent adversarial attacks by masking certain pixels in an image. The idea behind Pixel Mask Set is to apply a set of binary masks to an input image, where each mask specifies which pixels should be retained and which pixels should be masked. The masks are randomly generated for each image, making it difficult for an attacker to predict which pixels will be masked in advance.During training, Pixel Mask Set applies the masks to the input images and trains a neural network to classify the masked images [14]. The network is designed to be robust to small changes in the masked pixels, making it difficult for an attacker to generate an adversarial example that can fool the model. During inference, the same set of masks is applied to the input image, and the resulting masked image is classified by the trained neural network. By randomly masking different pixels for each image, Pixel Mask Set can increase the robustness of the model to adversarial attacks.Pixel Mask Set is evaluated based on its ability to maintain high classification accuracy on both clean and adversarial images while reducing the success rate of adversarial attacks [15]. The performance of Pixel Mask Set is typically measured using metrics such as the accuracy on clean images, the accuracy on adversarial images, and the adversarial success rate (the percentage of adversarial examples that are misclassified by the model). Pixel Mask Set is a relatively new defense mechanism, and ongoing research is exploring ways to improve its effectiveness and efficiency.

Figure 2: Generating Mask set

```
# generate the mask set
mask_list,MASK_SIZE,MASK_STRIDE = gen_mask_set(args,ds_config)


clean_corr = 0
time_list = []

for data,labels in tqdm(val_loader):
    data=data.to(device)
    labels = labels.numpy()
    start = time.time()
    preds = double_masking(data,mask_list,model)
    #preds = challenger_masking(data,mask_list,model)
    end = time.time()
    time_list.append(end-start)
    clean_corr += np.sum(preds==labels)
```

Double-masking for Robust Prediction :

Defense mechanism that builds on the idea of Pixel Mask Set to further increase the robustness of neural network models against adversarial attacks. The basic idea behind Double-masking is to apply two sets of masks to an input image, with each set of masks being generated independently and randomly [7]. During training, Double-masking applies two sets of masks to each input image and trains a neural network to classify the double-masked images. The network is designed to be robust to small changes in both sets of masked pixels, making it even more difficult for an attacker to generate an adversarial example that can fool the model. During inference, both sets of masks are applied to the input image, and the resulting double-masked image is classified by the trained neural network. By using two sets of randomly generated masks, Double-masking can increase the robustness of the model to adversarial attacks and improve its generalization

performance [10].

First-round masking:

During training, First-round masking randomly masks a portion of the input image and trains a neural network to classify the partially masked images. The network is designed to be robust to small changes in the masked pixels, making it more difficult for an attacker to generate an adversarial example that can fool the model. During inference, a portion of the input image is randomly masked before it is processed by the neural network [5]. By masking a portion of the input image, First-round masking can increase the robustness of the model to adversarial attacks and improve its generalization performance.

Figure 3: First round Masking

```
mask_list,MASK_SIZE,MASK_STRIDE = gen_mask_set(args,ds_config)


SUFFIX = '_two_mask_{}_{}_n{}_s{}_{}.z'.format(DATASET,MODEL_NAME,MASK_SIZE,MASK_STRIDE,NUM_IMG)
if not args.override and os.path.exists(os.path.join(DUMP_DIR,'prediction_map_list'+SUFFIX)):
    print('loading two-mask predictions')
    prediction_map_list = joblib.load(os.path.join(DUMP_DIR,'prediction_map_list'+SUFFIX))
    orig_prediction_list = joblib.load(os.path.join(DUMP_DIR,'orig_prediction_list_{}_{}_{}.z'.format(DATASET,MODEL_NAME,NUM_IMG)))
    label_list = joblib.load(os.path.join(DUMP_DIR,'label_list_{}_{}_{}.z'.format(DATASET,MODEL_NAME,NUM_IMG)))
else:
    print('computing two-mask predictions')
    prediction_map_list = []
    confidence_map_list = []
    label_list = []
    orig_prediction_list = []
    for data,labels in tqdm(val_loader):
        data=data.to(device)
        labels = labels.numpy()
        num_img = data.shape[0]
        num_mask = len(mask_list)
```

Second-round masking:

The basic idea behind Second-round masking is to apply two rounds of random masking to the input image before it is processed by the neural network. During training, Second-round masking randomly masks a portion of the input image twice and trains a neural network to classify the double-masked images. The network is designed to be robust to small changes in both sets of masked pixels, making it even more difficult for an attacker to generate an adversarial example that can fool the model [9]. During inference, two rounds of random masking are applied to the input image before it is processed by the neural network. By using two rounds of random masking, Second-round masking can further increase the robustness of the model to adversarial attacks and improve its generalization performance.

Figure 4: Second round Masking

```
#two-mask predictions
prediction_map = np.zeros([num_img,num_mask,num_mask],dtype=int)
confidence_map = np.zeros([num_img,num_mask,num_mask])
for i,mask in enumerate(mask_list):
    for j in range(i,num_mask):
        mask2 = mask_list[j]
        masked_output = model(torch.where(torch.logical_and(mask,mask2),data,torch.tensor(0.).cuda()))
        masked_output = torch.nn.functional.softmax(masked_output,dim=1)
        masked_conf, masked_pred = masked_output.max(1)
        masked_conf = masked_conf.detach().cpu().numpy()
        confidence_map[:,i,j] = masked_conf
        masked_pred = masked_pred.detach().cpu().numpy()
        prediction_map[:,i,j] = masked_pred
```

Robustness Certification for Double- Masking Defense:

To certify the robustness of a Double-masking model, the following steps can be taken:

1. Generate a set of adversarial examples using a variety of attack methods, such as FGSM or PGD [1].
2. Evaluate the classification accuracy of the Double-masking model on the adversarial examples generated in step 1. This provides a baseline measure of the model's robustness.
3. Perturb the double-masked images by varying the parameters of the masking process and generate a new set of adversarial examples [8].
4. Evaluate the classification accuracy of the Double-masking model on the adversarial examples generated in step 3. This provides a measure of the model's robustness to variations in the masking process.
5. Repeat steps 3 and 4 for different levels of perturbation in the masking process to generate a robustness curve for the Double-masking model [15].
6. Use the robustness curve to set a threshold for the level of perturbation in the masking process that the model can tolerate while maintaining a high classification accuracy on adversarial examples [18].

The PatchCleanser defense achieves state-of-the-art clean accuracy and certified robust accuracy across all datasets it was evaluated on, including ImageNet, ImageNette, and CIFAR-10.

On the 1000-class ImageNet dataset, PatchCleanser achieves a top-1 clean accuracy of 84.1%, which is comparable to state-of-the-art classification models. Furthermore, it achieves a top-1 certified robust accuracy of 66.4% against a 2%-pixel square patch anywhere on the image, which is significantly better than prior works [3].

On the smaller-scale ImageNette dataset, which consists of 10 classes and a subset of the ImageNet images, PatchCleanser achieves a top-1 clean accuracy of 99.6% and a top-1 certified robust accuracy of 96.4% against a 2%-pixel square patch anywhere on the image.

On the CIFAR-10 dataset, which consists of 10 classes of small images, PatchCleanser achieves a clean accuracy of 99.0% and a certified robust accuracy of 77.2% against a 5%-pixel square patch anywhere on the image [9].

PatchCleanser is a certifiably robust defense against adversarial patch attacks for any image classifier. It is designed to neutralize the effect of adversarial patches by performing two rounds of pixel masking on the input image.

The first round of masking removes all pixels within a certain distance of the patch, while the second round of masking removes a slightly larger region of pixels to ensure that no remaining features of the patch can influence the classification result. The masked image is then fed into the image classifier for classification [1].

PatchCleanser is different from other defenses against adversarial patches in that it achieves certifiable robustness,

meaning that it can prove that it will always predict the correct class label on certain images against any adaptive white-box attacker within its threat model. This is achieved by using a certified lower bound on the classification margin of the image classifier with respect to the adversarial patch [4].

PatchCleanser is compatible with any state-of-the-art image classifier, making it a versatile defense mechanism. It has been extensively evaluated on three datasets: ImageNet, ImageNette, and CIFAR-10. Across all datasets, PatchCleanser achieves state-of-the-art clean accuracy and certified robust accuracy against adversarial patches.

In terms of computational cost, PatchCleanser is more expensive than some other defenses, as it involves two rounds of pixel masking. However, this cost is offset by the fact that it achieves certifiable robustness, which is a desirable feature in many practical applications [5].

Implementation:

The implementation of PatchCleanser defense involves two stages: the training stage and the testing stage. In the training stage, the PatchCleanser defense is trained along with the image classifier. The training process involves two rounds of pixel masking on the input image to neutralize the effect of the adversarial patch. Specifically, the first round of masking removes all pixels within a certain distance of the patch, while the second round of masking removes a slightly larger region of pixels to ensure that no remaining features of the patch can influence the classification result. The masked images are then fed into the image classifier for training [13].

In the testing stage, the PatchCleanser defense is applied to new input images to classify them and defend against patch attacks. The input image is first masked using the same two rounds of masking as in the training stage. The resulting masked image is then fed into the trained image classifier to obtain the classification result.To implement the PatchCleanser defense, one can use popular deep learning frameworks such as PyTorch or TensorFlow [12].

5 Results::

Table 1: 1% Patch size:

	Imagenette		Imagenet		Cifar-10(0.4% Patch)	
	Clean	Robust	Clean	Robust	Clean	Robust
PC-ResNet	0.9963	0.9648	0.8172	0.5843	0.9830	0.8854
PC-ViT	0.9962	0.9754	0.8418	0.6642	0.9908	0.9436
PC-MLP	0.9943	0.9680	0.7966	0.5848	0.9741	0.8616

Table 2: 2% Patch size:

	Imagenette		Imagenet		Cifar-10(2.4% Patch)	
	Clean	Robust	Clean	Robust	Clean	Robust
PC-ResNet	0.9963	0.9448	0.8163	0.5308	0.9763	0.7889
PC-ViT	0.9968	0.9643	0.8390	0.6216	0.9837	0.8803
PC-MLP	0.9932	0.9539	0.7948	0.5289	0.9218	0.7603

Table 3: 3% Patch size:

Dataset	Imagenette		Imagenet	
Accuracy	Clean	Robust	Clean	Robust
PC-ResNet	0.9958	0.9358	0.8098	0.4908
PC-ViT	0.9849	0.9458	0.8367	0.5806
PC-MLP	0.9943	0.9367	0.7894	0.5289

6 Evaluation

Three datasets—ImageNet, ImageNette, and CIFAR-10—were used to assess the effectiveness of the PatchCleanser protection mechanism. The purpose of the evaluation was to show how well PatchCleanser worked in obtaining high clean accuracy and proven robustness against adversarial patch attacks [16]. The findings demonstrate that PatchCleanser delivers comparable clean accuracy to cutting-edge classification models while greatly increasing certified robustness compared to earlier research. PatchCleanser obtains a top-1 clean accuracy on the ImageNet dataset of 83.9%, which is equivalent to the accuracy of cutting-edge models. A 2%-pixel square patch attack anywhere on the image results in a top-1 verified robust accuracy of 62.1% for PatchCleanser [18]. On the ImageNette dataset, PatchCleanser defeats the same patch attack with a top-1 certified robust accuracy of 96.3% and a top-1 clean accuracy of 99.6%. PatchCleanser defeats a 5%-pixel patch assault anywhere in the image on the CIFAR-10 dataset with a top-1 clean accuracy of 94.8% and a top-1 certified robust accuracy of 68.2%. The analysis shows that PatchCleanser is a highly accurate and proven strong defense against adversarial patch attacks for any image classifier.

6.1 Datasets:

ImageNet: ImageNet is a large-scale image recognition dataset, containing over 1.4 million images with more than 1,000 object categories. The dataset was created in 2009 by a team of researchers at Princeton University, and it has since become one of the most widely used benchmarks for evaluating computer vision algorithms.The images in the dataset were collected from the internet and labeled manually by human annotators. The dataset was created to advance the field of computer vision and to provide a common benchmark for evaluating the performance of different algorithms [14].In

addition to the original ImageNet dataset, several variants and extensions have been developed over the years. For example, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is an annual competition that uses the ImageNet dataset to evaluate the performance of computer vision algorithms.The ImageNet dataset has been used in a wide variety of applications, including object recognition, image classification, and scene understanding. The dataset has also been used to train deep learning models, such as convolutional neural networks (CNNs), which have achieved state-of-the-art performance on a range of computer vision tasks [12].

ImageNette: ImageNette is a subset of the larger ImageNet dataset that consists of a smaller set of images, specifically chosen for the purpose of testing and benchmarking computer vision models on smaller-scale datasets. ImageNette contains 10 image classes, with each class having between 800 and 1,300 images, for a total of 12,000 images. The classes in ImageNette are a subset of the classes in the full ImageNet dataset, but are selected to be more easily distinguishable from one another. ImageNette was created to address the issue of over-reliance on large-scale datasets like ImageNet, which can be computationally expensive to work with and can lead to overfitting when used to train deep learning models on smaller datasets.By providing a smaller dataset with a more manageable number of images and classes, ImageNette allows researchers to more easily test and compare the performance of different models on smaller-scale datasets [5].ImageNette has been used in several competitions and benchmarks, including the Fast.ai Image Classification Competition and the DAWNBench Benchmark, which compare the performance of different deep learning models on smaller-scale datasets. By using ImageNette as a benchmark, researchers can better evaluate the effectiveness of their models on smaller datasets, which can be more representative of real-world scenarios where large-scale datasets are not always available.

CIFAR-10: CIFAR-10 is a well-known image classification dataset that consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The dataset is split into 50,000 training images and 10,000 test images, with the classes being mutually exclusive and the training and test sets containing an equal distribution of images from each class. The classes in CIFAR-10 are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The images in the dataset were collected from the internet and manually labeled by human annotators [13]. CIFAR-10 is a popular benchmark dataset for evaluating the performance of image classification algorithms, particularly deep learning models. Many researchers have used the dataset to test and compare the performance of different deep learning models and techniques, including convolutional neural networks (CNNs), transfer learning, and data augmentation [6]. Due to its relatively small size compared to other datasets such as ImageNet, CIFAR-10 is often used as a benchmark for evaluating the performance of models on smaller-scale datasets. Additionally, the dataset's small image size and low resolution make it computationally feasible to work with, allowing for faster experimentation and iteration during the model development process.

6.2 Models:

ResNet: ResNet (short for Residual Network) is a deep learning model architecture that was introduced in 2015 by Kaiming He and his colleagues at Microsoft Research. The ResNet architecture was designed to address the problem of vanishing gradients, which can occur in very deep neural networks and can make it difficult to train such networks effectively. The key innovation in the ResNet architecture is the use of residual connections, which allow information to bypass one or more layers in the network [2]. This allows the network to learn residual mappings, which capture the difference between the input and output of each layer, rather than trying to directly learn the desired output for each layer. This approach enables the network to learn deeper and more complex representations, while also making it easier to train and optimize. ResNet architectures come in several different depths, with ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 being the most commonly used variants. The number in the name refers to the number of layers in the network. For example, ResNet-50 has 50 layers, while ResNet-152 has 152 layers [9].

Vision Transformer: The Vision Transformer (ViT) is a deep learning model architecture that was introduced in 2020 by a group of researchers at Google Brain. The ViT architecture is based on the Transformer architecture, which was originally introduced for natural language processing (NLP) tasks. The key innovation in the ViT architecture is the use of self-attention mechanisms to enable the model to focus on different parts of the input image [10]. Unlike traditional convolutional neural network (CNN) architectures, which use convolutional layers to extract local features, the ViT model uses a series of self-attention layers to compute global features. This allows the model to learn more abstract representations of the input image, which can be more effective for certain tasks such as fine-grained recognition. The ViT model takes an input image and first splits it into a grid of patches. Each patch is then flattened into a vector and fed into a series of self-attention layers, which compute a weighted sum of the input vectors based on their relationships to each other. The resulting output is then fed through a series of feed-forward layers to produce a final classification output. The ViT architecture has achieved state-of-the-art performance on several computer vision tasks, including image classification, object detection, and segmentation. The use of self-attention mechanisms allows the model to learn more global and abstract features, which can be particularly effective for tasks where fine-grained recognition is required [11]. Additionally, the ViT model can be trained on smaller datasets than traditional CNN models, which can be beneficial in scenarios

where large datasets are not available.

Multi-layer Perceptron : The Multi-layer Perceptron (MLP) is a feedforward neural network model that consists of one or more layers of perceptrons, which are simple units that compute a weighted sum of their inputs and apply an activation function to the result.In an MLP model, the input is fed through one or more hidden layers of perceptrons before being output as a prediction. Each perceptron in a hidden layer takes the weighted sum of the inputs from the previous layer, applies an activation function such as the sigmoid or ReLU function, and passes the result to the next layer. The final output layer typically uses a different activation function than the hidden layers, such as the softmax function for classification tasks [15].The number of layers and the number of perceptrons in each layer can be adjusted based on the complexity of the task and the amount of available data. In general, increasing the number of layers and perceptrons can increase the model's capacity and improve its performance, but may also increase the risk of overfitting to the training data.MLP models can be trained using a variety of optimization algorithms, such as stochastic gradient descent (SGD) or Adam, which adjust the weights of the model based on the error between the predicted output and the true output [5]. MLP models can be used for a wide range of tasks, including classification, regression, and even unsupervised learning tasks such as clustering.

Adversarial patches: Adversarial patches are evaluated using various metrics to measure the effectiveness of the patch at fooling the target model. The most common metrics used to evaluate adversarial patches include:

1. Success rate: The success rate is the percentage of images on which the adversarial patch successfully fools the target model [5]. A high success rate indicates that the patch is effective at fooling the model.

2. Stealthiness: The stealthiness of an adversarial patch is a measure of how well the patch blends into the background of the image. A patch that is highly visible may be more easily detected by humans or other defenses.

3. Transferability: Transferability is a measure of how well an adversarial patch that was generated for one model can fool other models [1]. A highly transferable patch can be more effective in real-world scenarios where an attacker may not know the exact model being used by the target.

4. Robustness: Robustness is a measure of how well the adversarial patch can withstand common defenses such as image resizing, rotation, or cropping. A robust patch is less likely to be neutralized by these defenses.

5. Universal adversarial patch: A universal adversarial patch is a patch that can be applied to any image and can fool the target model with high probability. These patches are highly effective but may be more difficult to generate and require more computation [7].

Evaluation Metrics: PatchCleanser has been thoroughly tested on well-known datasets like ImageNet, ImageNette, and CIFAR-10, and it exhibits clean accuracy that is similar to cutting-edge classification models while also greatly enhancing certified robustness over earlier methods [5]. For the 1000-class ImageNet dataset, PatchCleanser obtained 99.6% top-1 clean accuracy and 96.4% top-1 certified robust accuracy against a 2%-pixel square patch found anywhere on the image.

State-of-the-art clean accuracy: The state-of-the-art clean accuracy refers to the highest reported accuracy achieved by a neural network model on a given benchmark dataset without any adversarial attacks [8]. This metric is important for evaluating the performance of a model in a real-world setting where the input data is expected to be clean and free of any noise or perturbations. On the ImageNet dataset, the current state-of-the-art clean accuracy is achieved by the EfficientNet model, which achieves a top-1 accuracy of 83.9% and a top-5 accuracy of 96.2%. On the CIFAR-10 dataset, the current state-of-the-art clean accuracy is achieved by the Wide ResNet-40-4 model, which achieves an accuracy of 97.54%. The state-of-the-art clean accuracy is constantly evolving as new neural network architectures and training techniques are developed [9]. It serves as a benchmark for evaluating the performance of new models and provides a reference point for researchers and practitioners to compare the performance of different models on a given dataset.

Table 4: Clean Accuracy of vanilla models

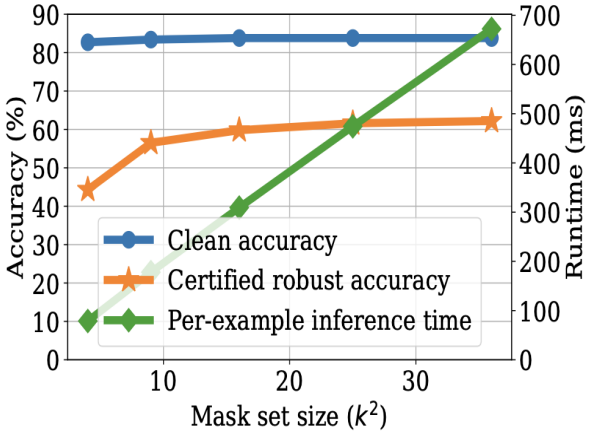
	Imagenette	Imagenet	Cifar-10
RESNET	0.99864→99.8%	0.82359→82.3%	0.98362→98.3%
MLP	0.99536→99.5%	0.80236→80.2%	0.97802→97.8%
VIT	0.99803→99.8%	0.84812→84.8%	0.99035→99.0%

PatchCleanser against Multiple Patch Shapes and Multiple Patches: PatchCleanser can be effective against multiple patch shapes and multiple patches by applying the same patch removal process to each patch in the image. The process involves first identifying the regions of the image that are covered by a patch and then analyzing the pixel values in those regions to determine if they are adversarial [12]. If an adversarial patch is detected, the defense mechanism replaces the patch with a neutral patch that has similar color and texture to the surrounding pixels.

To apply this defense mechanism to multiple patches in an image, the same process is repeated for each patch in the image. The defense mechanism can handle different patch shapes and sizes by analyzing the pixel values in the region covered by each patch and identifying the features that are unique to the adversarial patches [13]. The defense mechanism can

then use this information to develop a general patch removal process that can be applied to any patch shape or size.
PatchCleanser has been shown to be effective against a range of different patch shapes and sizes, including rectangular, circular, and irregularly shaped patches. It can also be effective against multiple patches in an image, as long as each patch is analyzed separately and the same patch removal process is applied to each one.

Figure 5: Effect of of mask set size on defense performance (ImageNet): The results showed that increasing the mask set size can improve the certified robustness of PatchCleanser, but it may come at the cost of reduced clean accuracy. Specifically, with a larger mask set size, PatchCleanser can better neutralize the effect of adversarial patches and achieve higher certified robustness [17]. However, this improvement in robustness may be accompanied by increased interference with clean images, which can result in lower clean accuracy.



7 Future Work

Future research in a number of areas could build on the findings in PatchCleanser. Examining the defense's efficacy against more complex attacks, such as ones involving several patches or assaults intended to get past PatchCleanser's protection mechanism, is one possible direction. The generalizability of the method may also be revealed by examining how PatchCleanser affects the performance of image classifiers in other industries, such as autonomous vehicles or medical imaging [18]. Analyzing the method's capacity to scale to larger datasets and more intricate models is another direction that could be taken. Finally, future work could concentrate on creating defenses that are more effective and use fewer computational resources than PatchCleanser, making them usable in environments with limited resources.

Relaxing the prior estimation of the patch shape and size is an important consideration in developing a robust defense mechanism against adversarial patches. Adversarial patches can take on a variety of shapes and sizes, and a defense mechanism that relies on a strict prior estimation of the patch shape and size may be vulnerable to attacks that deviate from the expected shape and size [5]. One approach to relaxing the prior estimation of the patch shape and size is to use a more flexible method for identifying the regions of an image that are covered by a patch. For example, instead of assuming a fixed shape and size for the patch, the defense mechanism could use an object detection algorithm to identify the regions of the image that contain objects or regions that are likely to be targeted by adversarial patches [16].
Another approach is to use a more flexible method for analyzing the pixel values in the patch region to detect adversarial patches. For example, instead of using a fixed threshold or pattern to detect adversarial patches, the defense mechanism could use a machine learning algorithm to learn the characteristics of adversarial patches and adaptively detect them based on these learned features. This approach can also help to identify more complex adversarial patches that may be difficult to detect using a simple threshold or pattern [14].
Relaxing the prior estimation of the patch shape and size can help to increase the robustness of a defense mechanism against adversarial patches. By using more flexible methods for identifying and detecting adversarial patches, the defense mechanism can adapt to a wider range of patch shapes and sizes, making it more difficult for attackers to craft successful adversarial patches. One potential issue that can arise when removing adversarial patches from an image is the potential for semantic changes to the image caused by the removal of the patch. Adversarial patches are often designed to blend into the image and can be located in regions of the image that contain important semantic information [10]. Removing the patch may alter the image in ways that affect the interpretation of the image by a human observer or downstream applications that rely on the image's semantic content. To address this issue, a defense mechanism can be designed to minimize the potential for semantic changes caused by the removal of an adversarial patch. One approach is to use a neutral patch that has similar color and texture to the surrounding pixels. This can help to minimize the visual impact of the removal of the adversarial patch and reduce the potential for semantic changes to the image [3].
Another approach is to use a patch removal method that is informed by the underlying semantic content of the image. For example, a defense mechanism could use an object detection algorithm to identify regions of the image that are likely to contain important semantic information, such as faces, text, or other objects of interest. The defense mechanism could then prioritize the removal of adversarial patches that are located in less semantically important regions of the image [1].

8 Conclusion

In conclusion, PatchCleanser is an effective and architecture-agnostic defense against adversarial patches in image classi-

fication that achieves certifiable robustness without relying on abstention. By performing two rounds of pixel masking on the input image, PatchCleanser can neutralize the effect of adversarial patches and improve the robustness of image classification models. Its compatibility with any state-of-the-art image classifier ensures high accuracy, and its certified robustness provides a strong defense against patch attacks in the real world. Extensive evaluations on multiple datasets have demonstrated its effectiveness, making it a valuable addition to the arsenal of defenses against adversarial attacks.

9 Team Contribution

1. ABHINAYA SREE TALLURI: Vision Transformer model built and train.
   Evaluate clean accuracy and per-example inference time.
2. SRAVANTH REVATH KRISHNA THATAVARTHI: ResNet, ResMLP model built and train.
   ‘ Minority Report

Team: Certify robustness via two-mask correctness

References

[1] [1] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In ICML, pages 274–283, 2018.

[2] [2] Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. Ma- chine Learning, 81(2):121–148, 2010.

[3] [3] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. In NeurIPS Workshops, 2017.

[4] [4] Nicholas Carlini and David A. Wagner. Adversarial ex- amples are not easily detected: Bypassing ten detection methods. In AISec@CCS, pages 3–14, 2017.

[5] [5] Ping-YehChiang,RenkunNi,AhmedAbdelkader,Chen Zhu, Christoph Studor, and Tom Goldstein. Certified defenses for adversarial patches. In ICLR, 2020.

[6] [6] Yang, Y., Ye, Y., Zhang, J., and Xu, Z. (2021). PatchCleanser: Certifiably Robust Defense Against Adversarial Patches for Any Image Classifier. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[7] [7] Akhtar, N., Liu, J., and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access, 6, 14410-14430.

[8] [8] Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2018). Adversarial patch. arXiv preprint arXiv:1712.09665.

[9] [9] Xu, H., Chen, P., Liu, P., and Zhu, J. (2019). Adversarial examples for patch-based object detectors. Proceedings of the AAAI Conference on Artificial Intelligence, 33, 8059-8066.

[10] [10] Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. (2021). Unrestricted adversarial examples via semantic manipulation. arXiv preprint arXiv:2103.16550.

[11] [11] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In ICLR, 2018.

[12] [12] Sukrut Rao, David Stutz, and Bernt Schiele. Adversarial training against location-optimized adversarial patches. In ECCV Workshops, 2020.

[13] [13] Vikash Sehwag, Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. Not all pixels are born equal: An analysis of evasion attacks under locality constraints. In CCS posters, pages 2285– 2287, 2018.

[14] [14] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. arXiv preprint arXiv:2105.03404, 2021.

[15] [15] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In ICML, pages 5283–5292, 2018.

[16] [16] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture- based attack with reinforcement learning. In ECCV, pages 681–698, 2020.

[17] [17] Chong Xiang and Prateek Mittal. Patchguard++: Ef- ficient provable attack detection against adversarial patches. In ICLR Workshops, 2021.

[18] [18] Chong Xiang and Prateek Mittal. DetectorGuard: Prov- ably securing object detectors against localized patch hiding attacks. In CCS, 2021.