Minority Reports Defense: Defending Against Adversarial Patches

Michael McCoyd[1 [0000−0003−4349−4648]], Won Park[1,2 [0000−0002−3032−0748]], Steven Chen[1 [2222−−3333−4444−5555]], Neil Shah[1 [2222−−3333−4444−5555]], Ryan Roggenkemper[1 [2222−−3333−4444−5555]], Minjune Hwang[1 [0000−0002−3697−8444]], Jason Xinyu Liu[1 [0000−0001−7732−3666]], and David Wagner[1 [2222−−3333−4444−5555]]

[1] University of California, Berkeley, Berkeley CA 94720, USA
mmccoyd,daw@cs.berkeley.com
scchen,neilshah430,rroggenkemper,mjhwang,xinyuliu@berkeley.edu
[2] University of Michigan, Ann Arbor, MI 48109, USA wonpark@umich.edu

Abstract. Deep learning image classification is widely used yet is vulnerable to adversarial attack, which can change the computer classification without changing how humans classify the image. This is possible even if the attacker changes just a small patch of the image. We propose a defense against patch attacks based on partially occluding the image around each candidate patch location, so that a few occlusions each completely hide the patch. We demonstrate on CIFAR-10, Fashion MNIST, and MNIST that our defense provides certified security against patch attacks of a certain size. For CIFAR-10 and a 5×5 patch, we can provide certify accuracy for 43.8% of images, at a cost of only 1.6% in clean image accuracy compared to the architecture we defend or a cost of 0.1% compared to our training of that architecture, and a 0.2% false positive rate.

Keywords: Adversarial Machine Learning · Adversarial Patch · Partial Occlusions Ensemble Defense.

1 Introduction

Deep learning image classification is widely used yet is vulnerable to adversarial attack, which can change the computer classification without changing how humans classify the image. An attacker with knowledge of a neural network model can construct, from any normal image x, an *adversarial example* $x^\star$ that looks to humans like x but that the model classifies differently from the normal image [SZS+14], [GSS15], [HJN+11], [CW17].

Recently, researchers have proposed the *adversarial patch* attack [BMR+17], [KZG18], where the attacker changes just a limited rectangular region of the image, for example, by placing a sticker over a road sign or other object. Others have expanded on the vulnerability to this type of attack [EEF+17], [TRG19], [XZL+19]. In this paper, we propose a defense against this attack.

The idea of our defense is to occlude part of the image and then classify the occluded image. First, we train a classifier that properly classifies occluded

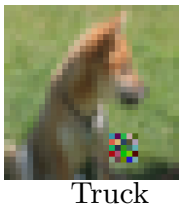
images. Then, if we knew the location of the adversarial patch, we could occlude that region of the image (e.g., overwriting it with a uniform grey rectangle) and apply the classifier to the occluded image. This would defend against patch attacks, as the attacker's contribution is completely overwritten and the input to the classifier (the occluded image) cannot be affected by the attacker in any way.

In practice, we do not know the location of the adversarial patch, so a more sophisticated defense is needed. Our approach works by occluding an area larger than the maximum patch size and striding the occlude area across the image, making an occluded prediction at each stride. We then analyze the classifier's predictions on these occluded images. If the occlusion region is sufficiently larger than the adversarial patch, several of the occluded images will completely obscure the adversarial patch and thus the classifier's prediction on those images will be unaffected by the adversary and should match the correct label. Thus, we expect the correct label to appear multiple times among the predictions from occluded images. We show how to use this redundancy to detect adversarial patch attacks. We call our scheme the minority reports defense because no matter where the patch is located, there will always be a minority of predictions that cannot be influenced by the attacker and vote for the correct label.

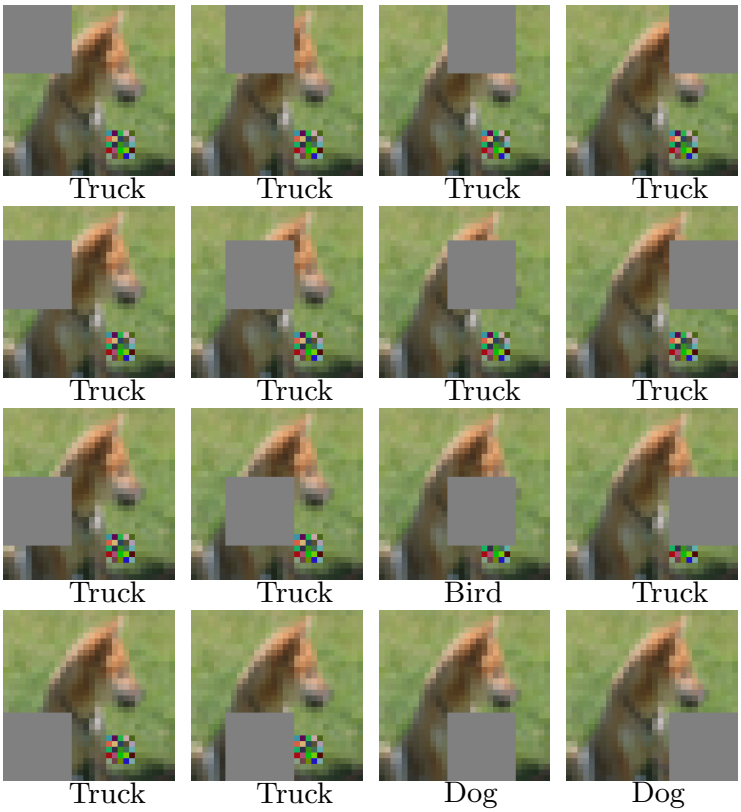
Figure 1 illustrates our defense. We take the input image (Figure 1a) and construct a grid of partially occluded images (Figure 1b) with occlusions at different locations, chosen so that any attack will be occluded in a cluster of several predictions. We then apply the classifier to each occluded image to obtain a grid of predictions. When under attack, we can expect most predictions to differ from the true label, but there will always be a cluster of locations where the adversarial patch is fully obscured, and thus the labels are all expected to agree with the true label; in Figure 1, the 3rd and 4th images in the 4th row obscure the adversarial patch and thus vote for the true label. Our defense analyzes the grid of predicted labels to detect this pattern. If there is a cluster of predictions that all match each other but are in the minority for the prediction grid overall, then this suggests an attack. Figure 2 visualizes the prediction grid for a benign image (on the left) and a malicious image containing an undefended adversarial patch (on the right).

We evaluate our scheme on the CIFAR-10 [KH09], Fashion MNIST [XRV17] and MNIST [LBBH98] datasets with a stride of one. We show that our defense does not harm accuracy much. We also evaluate its security against adaptive attacks. In particular, we show how to bound the success of any possible attack on a given image, and using this, we are able to demonstrate certified security for a large fraction of images. In particular, we are able to prove a security theorem: for a large fraction of images in the validation set, we can prove that no patch attack will succeed, no matter where the patch is placed or how the patch is modified, so long as the size of the patch is limited. In summary, our contributions in this paper are:

– We quantify the vulnerability of undefended networks for Fashion MNIST and MNIST against patch attacks with patches of different sizes (§2.2).

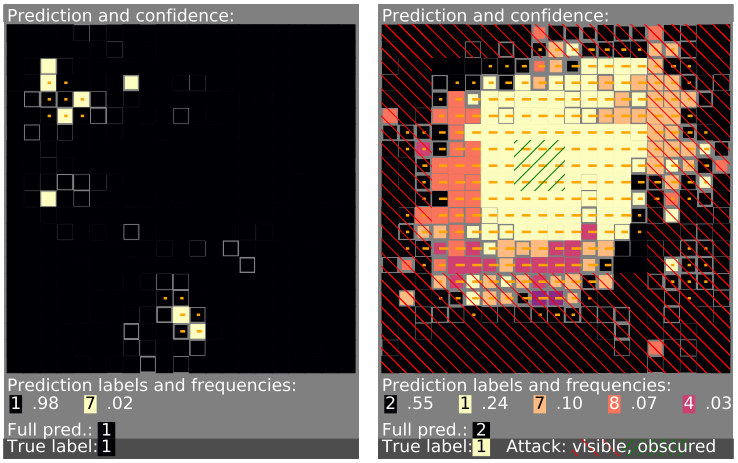


(a) An attack image: a picture of a dog with a malicious 5×5 sticker that causes a standard model to classify it as a truck.



(b) We occlude part of the image with a grey square, then classify these occluded images. Here the 3rd and 4th predictions in the 4th row will be unaffected by this attack. Our actual defense ensures that any attack will be fully occluded by a 3×3 grid of predictions, instead of the 1×2 grid shown here.

Fig. 1: Our scheme works by occluding different portions of the image and analyzing the predictions made by the classifier on these occluded images.



(a) Benign (b) Attack

Fig. 2: Prediction grids for a benign image (left) and an undefended attack image (right). Each cell in the grid is colored based on the classifier's prediction when fed an image obscured at that position in the grid. A cluster of identical minority predictions, as seen in the right image, suggests an attack. In the attack image on the right, green hashes mark the nine predictions where the adversarial patch was fully occluded.

– We propose a novel method for detecting patch attacks, based on differently occluded views of the input image (§3).
– We provide a worst-case analysis of security against adaptive attacks for CIFAR-10, Fashion MNIST, and MNIST (§4 and §6).

2 Patch Attack

Patch attacks [BMR+17] work by replacing a small part of the image with something of the attacker's choosing, e.g., by placing a small sticker on an object or road sign. Figure 1a shows a patch attack. Patch attacks represent a practical method of executing an attack in the physical world. Digital images can be manipulated throughout the entire scene they present, yet this is impractical in a physical, not digital, scene. It is far more practical to add an attacker-controlled object to part of the scene.

As a simple, non-malicious example, it is not uncommon to see stickers on road signs in the real world, without preventing humans from understanding the signs or prompting the patch's immediate removal.

2.1 Attack Model

We assume the attacker knows everything the defender knows: the architecture, weights, training data, and algorithm of all models and methods used by the defender. The attacker may place a 'compact' patch anywhere within the digital image and arbitrarily modify all pixels within the patch to any values in the pixel range. For simplicity, we restrict the attacker to a patch contained in an $n \times n$ square area for some n, n being a measure of the attacker's lack of stealth, and 'compact' meaning square.

The size of the adversarial patch that can be defended against can be thought of as similar to the size of an adversarial example L2 perturbation that can be defended against. Certainly, the attacker could make a larger change, but at some point, the change either becomes very obvious or changes the meaning of the image for humans. Thus crossing the fussy boundary from being an adversarial, stealthy, attack to being an image of something completely different to humans – thus no longer adversarial as described in §1

2.2 Patch Sizes

We first study how large of a patch is needed to successfully attack undefended Fashion MNIST and MNIST. We test multiple patch sizes and measure the attacker's success rate for each patch size.

Setup We conduct a targeted attack against standard Fashion MNIST and MNIST models from §6. We attack the first 300 validation images for Fashion MNIST and the first 100 validation images for MNIST. We report the fraction of images for which we can successfully mount a patch attack. For each image, we select a target label by choosing randomly among the classes that are least likely, according to the softmax outputs of the classifier (namely, we find the least likely class, identify all classes whose confidence is within 0.1% of the least likely, and select the target class uniformly at random among this set). That target is used for all attacks on that image. For each base image and its chosen target class, we enumerate all possible patch positions and try at each position to find an attack patch at that position.

Attack Algorithm To generate patch attacks, we iterate over all possible locations for the patch and use a projected gradient descent (PGD) attack for each location. We consider the attack a success if we find any location where we can place a patch that changes the model's prediction to the target label. The resulting adversarial patch is specific to one specific image and one specific location.

The standard PGD attack uses a constant step size, but we found it was more effective to use a schedule that varies the step size among iterations. In our experiments, a cyclic learning rate was more effective than a constant step size or an exponential decay rate, so we used it in all experiments. We used a cyclic learning rate with ten steps per cycle, with step sizes from 0.002 to 0.3, for a maximum of 150 steps. We stopped early at the end of a cycle if the attack

achieved confidence 0.6 or higher for the target class, or if the confidence had not improved by at least 0.002 in the last 20 steps from the best so far. For each image, we attacked in parallel across all possible patch locations.

Results For our MNIST model, a 6×6 patch is large enough to attack 45% of the images successfully. The success rate for 4×4 patches was 19%, and for 8×8 patches, 80%. When an image can be attacked, there are often many possible locations where an adversarial patch can be placed: for a 6×6 patch, out of all images where a patch attack is possible, there were, on average, 41 different positions where the patch can be placed.

For our Fashion MNIST model, the success rate for patch attacks was as follows: 4×4 patch: 27% success, 5×5 patch: 50% success, 6×6 patch: 60% success.

These results indicate that, on MNIST, an attacker needs to control a 6×6 patch to have close to a 50% chance of success, while a 5×5 patch is large enough for Fashion MNIST, occupying 5% and 3% of the images respectively.

We use 5×5 patches for CIFAR-10, Fashion MNIST and MNIST, as that size is used by recent work [CNA+20].

3 Our Defense

The basic idea of the minority reports defense is to occlude part of the image and classify the resulting image. If the occlusion completely covers the adversarial patch, then the attacker will be unable to influence the classifier's prediction. We don't know where the adversarial patch might be located, so we stride the occlusion area across the image. Because we use an occlusion area sufficiently larger than the adversarial patch, no matter where the adversarial patch is placed, there should be a cluster of occlusion positions that all yield the same prediction.

3.1 Occlusion Training

As our defense will internally use partial occlusions of the image it is given, we train, or retrain, with occluded images. Each time an image is presented in training, a randomly placed $n \times n$ square is occluded, and the model receives the occluded image. This is similar to cutout training from Devries et al. [DT17], who used occlusion as a regularizer. The difference in our training is that the occlusion is the size we will use in our defense. We also internally provide the model an additional input of a sparsity mask that indicates which pixels are occluded.

For instance, the input to an MNIST model is an image, with dimensions $28 \times 28 \times 1$, and a mask, with dimensions $28 \times 28 \times 1$. The image has its normal channels, and the mask has one channel. In the mask, a 0 indicates an occluded position, and a 1 a non-occluded position.

If a model already predicts accurately with a random partial occlusion of the size we use, there is no need to retrain or modify it, it can just be wrapped in our defense as described in the following sections.

To better handle the missing pixels, we modify the architectures we test by replacing convolutions with sparsity invariant convolutions [USS+17]. If the mask indicates no occlusions, the sparsity invariant convolutions behave as normal convolutions, but when occlusions are indicated, the occluded pixels are handled better.

Training on occluded images appears to have only a small change on the accuracy of the inner model on non-occluded images, see §A.

3.2 Creating a Prediction Grid

At evaluation time, our defense's first step is to generate a *prediction grid* as follows. We describe the simpler case of low-resolution images here, leaving the larger stride for higher resolution images to §5. For defending MNIST images against a 5×5 adversarial patch, we use a 7×7 occlusion region. We slide the 7×7 occlusion region over the 28×28 image with a stride of one pixel, yielding 26×26 possible locations for the occlusion region. This ensures any patch is covered by nine occlude areas, even a patch at the image edge, $26 = (28-(7-1))+2+2$. The prediction grid is a 26×26 array that records, for each location, the classifier's output. At each location, we mask out the corresponding occlusion region of the image, classify the occluded image, obtain the confidence scores from the classifier's softmax layer and record that in the corresponding cell of the prediction grid. Cell (i, j) of the prediction grid contains the confidence scores for all 10 classes when the pixels in the square $(i-2, j-2), \ldots, (i+5, j+5)$ of the image are masked out.

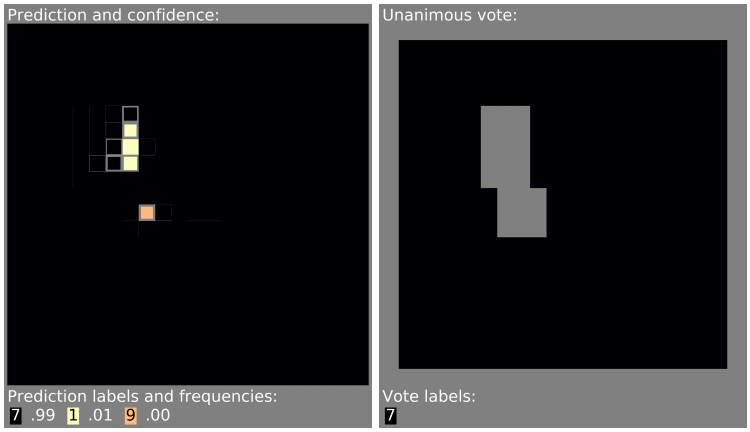
We visualize the pattern of occlusions in Figure 1b, though with a large stride for illustration. A stride of one on MNIST produces prediction grids such as figure 2 and figures 3a and 3c.

If the image contains an adversarial patch centered at location (i, j), then obscuring at each of the 9 locations centered at $(i-1, j-1), \ldots, (i+1, j+1)$ yields nine images where the adversarial patch has been completely overwritten, and the predictions in those cells of the prediction grid are completely unaffected by the attacker. If the classifier is sufficiently accurate on occluded images, we can hope that all of those 9 predictions match the true label. Thus, within the prediction grid, we can expect to see a 3×3 region where the predictions are uninfluenced by the attacker and (hopefully) all agree with each other. Our defense takes advantage of this fact.

3.3 Detection

In a benign image, typically, every cell in the prediction grid predicts for the same label. In contrast, in a malicious image, we expect there will be a 3×3 region in the prediction grid (where the adversarial patch is obscured) that predicts a single label and some or all of the rest of the prediction grid will have a different prediction. We use this to detect attacks.

In our simplest defense, we look at all 3×3 regions in the prediction grid that vote unanimously for the same label (i.e., all 9 cells yield the same classification).



(a) Scattered minority predictions (b) Vote grid: benign

(c) Cluster of minority predictions (d) Vote grid: attack

Fig. 3: In (a) and (c), we show the prediction grids for two benign images. (b) and (d) show the corresponding vote grids. We must decide if the minority votes (yellow) are benign errors or what remains of the truth after an attack has influenced the other predictions. Unanimous voting classifies the top example as benign and the bottom as an attack.

If there are two different labels that both have a 3×3 unanimous vote, then we raise an alarm and treat this as a malicious image.

Equivalently, we categorize each 3×3 region within the prediction grid as either unanimously voting for a class (if all 9 cells in that region vote for that class) or abstaining (if they don't all agree). We construct a 24×24 *voting grid* recording these votes. If the voting grid consists of solely a single class and abstentions, then we treat the image as benign, and we use that class as the final

prediction of our scheme. Otherwise, if the voting grid contains more than one class, we treat it as malicious.

The idea behind this defense is twofold. First, in a benign image, we expect it to be rare for any 3×3 region in the prediction grid to vote unanimously for an incorrect class: that would require the classifier to be consistently wrong on 9 occluded images. Therefore, the voting grid for benign images will likely contain only the correct class and abstentions. Second, for a malicious image, no matter where the adversarial patch is placed, there will be a 3×3 region in the prediction grid that is uninfluenced by the attack and thus can be expected to vote unanimously for the true class. This means that the voting grid for malicious images will likely contain the correct class at least once. This places the attacker in an impossible bind: if the attack causes any other class to appear in the voting grid, the attack will be detected, but if it does not, then our scheme will classify the image correctly. Either way, the defender wins.

We can formulate our defense mathematically as follows. Let x denote an image, $m_{i,j}$ denote the mask that occludes pixels in $[i-2, i+5] \times [j-2, j+5]$, and $x \odot m_{i,j}$ denote the result of masking image x with mask $m_{i,j}$. Then the prediction grid p is constructed as

$$p_{i,j} = C(x \odot m_{i,j}, m_{i,j}), \tag{1}$$

where the classifier C outputs a vector of confidence scores. The voting grid is defined as

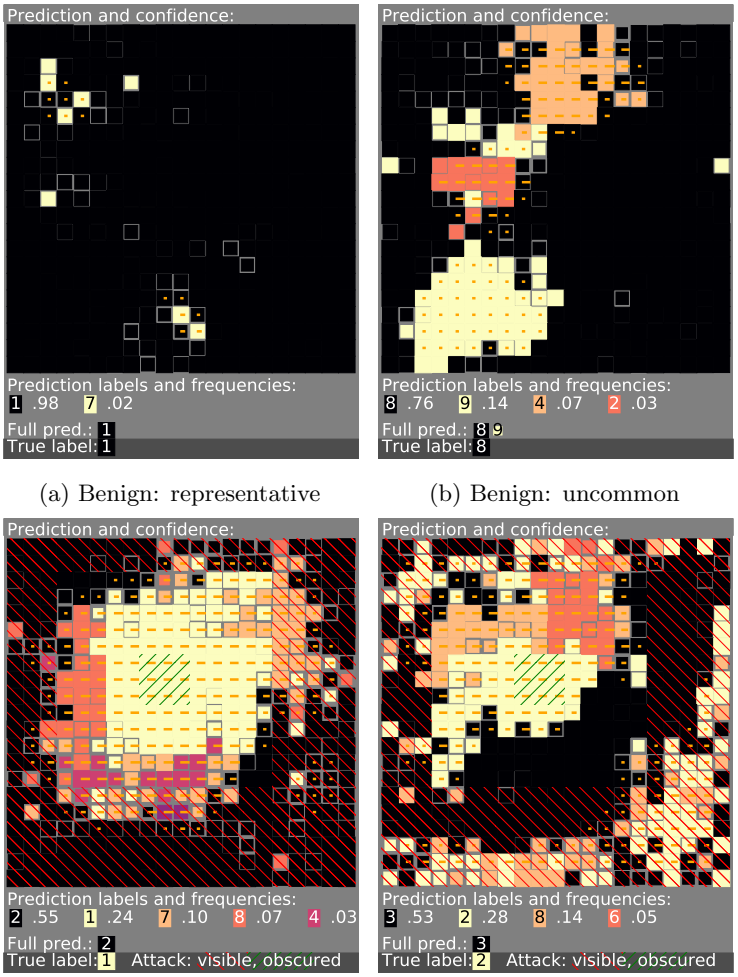
$$v_{i,j} = \begin{cases} c & \text{if } c = \arg\max_{c'} p_{i+u,j+v,c'} \ \forall u, v \in \{0,1,2\} \\ \text{␣} & \text{otherwise.} \end{cases} \tag{2}$$

If there exists a single class c such that $v_{i,j} = c$ or $v_{i,j} = \text{␣}$ for all i, j, then our scheme treats the image as benign and outputs the class c; otherwise, our scheme treats the image as malicious.

We illustrate how the defense works with two examples. For instance, if the prediction grid is as shown in figure 3a, then it yields the voting grid in figure 3b. This will be treated as benign, with classification 7. We show another example of a prediction grid in figure 3c and the resulting voting grid in figure 3d. This image will be treated as malicious, and our scheme will decline to classify it. In particular, it is possible that the true label is 5, but an adversarial patch was placed in the upper-left that caused most of the classifications to be shifted to 3, except for a few cases where the patch was partly or wholly obscured. It is, of course, also possible that the image was benign, and a cluster of classification errors caused this pattern, which is the case here.

3.4 Visualization

To give some intuition, we visualize a few sample prediction grids in figure 4. The 26×26 prediction grid is displayed as a Hinton diagram with 26×26 squares. The color of each square indicates which class had the highest confidence at that



(a) Benign: representative

(b) Benign: uncommon

(c) Attack

(d) Attack

Fig. 4: Representative prediction grids for benign and undefended attack MNIST images. Color indicates the arg max label for that occlusion position, and confidence is indicated by how much of the square is filled. We show at the bottom of each figure a legend indicating which class each color corresponds to and its frequency in the prediction grid; we also show the top prediction and confidence if no pixels are occluded. For attack images, green hashes show the 3×3 grid of predictions that completely occlude the attack; red hashes show the predictions that do not occlude the attack at all. The hashes are not part of our defense, merely an aid for the reader. (The short orange bars are from a detection method that compares with the non-occluded prediction.)

location in the prediction grid (i.e., the class predicted by the classifier). The size of each square is proportional to the confidence of that class.

We show a representative example from each of four different common cases that we have seen:

(a) Most benign images have a prediction grid that predicts all for the same label or has just scattered minority predictions and looks like case (a). The predictions almost always agree with the true label for almost all positions of the occlusion region. However, there are a few locations that, when occluded, cause classification errors (non-black squares). These will be correctly classified and treated as benign by our scheme.
(b) A few benign images have prediction grids that are noisier and contain large clusters of incorrect predictions in the prediction grid. These will be (incorrectly) categorized as malicious by our scheme, i.e., they will cause a false positive.
(c) We show the prediction grid resulting from a typical attack image, with an adversarial patch placed near the center of the image. The green cross-hatching represents the locations that completely occlude the adversarial patch. Those locations in the prediction grid, as well as some other locations in a broader ring around this, vote unanimously for the true label (`1`). Occlusion regions placed elsewhere fail to occlude the adversarial patch and cause the classifier to misclassify the image as the attacker's target class (`2`). Our scheme correctly recognizes this as malicious, because the voting grid contains both unanimous votes for `1` and for `2`.
(d) Other attack images have even more noise outside the fully occluded area. These, too, are correctly recognized as malicious because the voting grid contains unanimous votes for multiple labels, here 3, 2, and 6.

3.5 The Full Minority Reports Defense

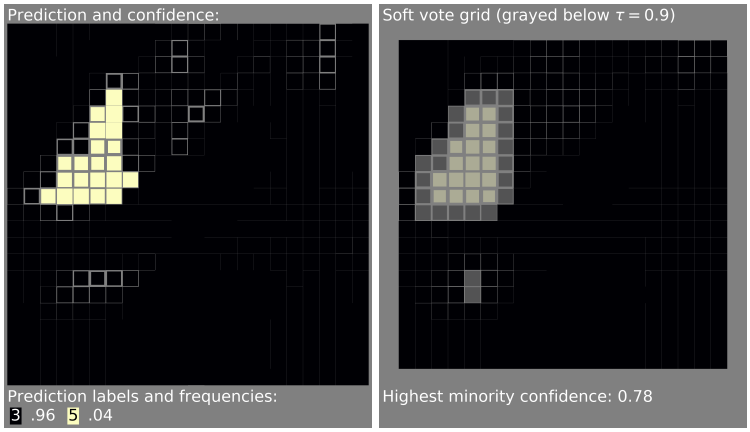
We found that the above defense can be improved by incorporating two refinements: (a) using soft agreement instead of hard unanimity, and (b) tolerating outliers.

First, instead of checking whether a 3×3 region in the prediction grid votes unanimously for the same label, we check whether the confidence for that label averaged over the region exceeds some threshold. For instance, with a 90% threshold, if the confidence scores for class c within that 3×3 region average to 0.9 or larger, then we'd record a vote for c in the voting grid; if no class exceeds the threshold, then we record an abstention.

Second, when computing the average, we discard the lowest score before computing the average. This allows us to tolerate a single outlier when checking for agreement in a 3×3 region.

Mathematically, we fix a threshold τ, and then form the voting grid as

$$v_{i,j} = \begin{cases} c & \text{if } \operatorname{avg}(\{p_{i+u,j+v,c}\ \forall u, v \in \{0,1,2\}\}) \geq \tau \\ \textvisiblespace & \text{otherwise.} \end{cases} \tag{3}$$



(a) Cluster of minority predictions (b) Vote grid: benign

Fig. 5: Our full defense on the benign prediction grid from figure 3c, with $\tau = 0.9$ classifying as benign (b). A sticker under any of the nonvoting areas would be undetected. A sticker in the lower right, when occluded, would leave in (a) the confident remains of the original prediction, and be classified as an attack.

Here we define $\text{avg}(S)$ to be the average of $S \setminus \{\min S\}$, i.e., the average of all but the lowest score in the multiset S.

The threshold τ is a hyper-parameter that can be used to control the trade-off between false positives and false negatives. Increasing τ reduces the number of false positives, but also risks failing to detect some attacks; decreasing τ increases detection power, at the cost of increasing the false positive rate.

The size of the occlusion region is another hyper-parameter of our defense. In our experiments, we always chose an occlusion region that is two pixels larger than the largest adversarial patch we seek to defend. Thus our occlusion region will be 7×7, and we provide certified results against adversarial patches up to 5×5 in size.

We visualize the operation of our final defense in figure 5.

4 Security Evaluation

One benefit of our design is that it enables us to guarantee the security of our scheme on some images. This provides a stronger result than evaluating against a specific adaptive attack. Were we to rely on evaluation against some adaptive attack, an adversary might be smarter than our adaptive attack and achieve a higher attack success rate. Instead, our certified result provides a guarantee that can not be beaten by any adaptive attack. We describe our certified security analysis in this section.

The core observation is: if the adversarial patch is completely occluded, then the adversary cannot have any influence on the prediction made by the classifier on the corresponding occluded image. For certified security, we make a very conservative assumption: we assume that the adversary might be able to completely control the classifier's prediction for all other occluded images (i.e., where the patch is only partly occluded or is not occluded at all). This assumption lets us make a worst-case analysis of whether the classification of a particular image could change in the presence of an adversarial patch of a particular size.

Notice that wherever the sticker is placed, there will be a 3×3 grid in the prediction grid that is unaffected by the sticker. (This is because with a stride of one, we use an occlusion region that is 2 pixels larger than the maximum possible sticker size.) It follows that there will be some cell in the voting grid that is not changed by the sticker.

If the voting grid for an image x is completely filled with votes for a single class c, with no abstentions, then any image x' that differs by introduction of a single sticker will either be classified by our defense as class c or will be detected by our defense as malicious. (This follows because at least one element of the voting grid is unaffected by the sticker, so at least one element of the voting grid for x' will vote for c. If no other class appears in the voting grid, then our defense will classify x' as class c; if some other class appears, then our defense will treat x' as malicious.) Thus, such images can be certified safe—there is no way to attack them without being detected. If the prediction is also correct, we classify the image as certified accurate.

In contrast, if the voting grid has even one region that does not vote or votes as the attacker would like, then our conservative analysis is forced to assume that it might be possible to attack the image: the attacker can place a sticker at that location, potentially changing all the other regions' votes, and thereby escape detection.

We evaluate the security of our scheme by measuring the fraction of images that can be certified safe and certified accurate, according to the conservative analysis above.

5 Higher Resolution Images

For higher resolution images, increasing the stride and pixel size of the occlude area lets us manage the cost of the prediction grid. For a patch of size $p \times p$ pixels and a stride of s pixels, an occlude area of $(p + 2s) \times (p + 2s)$ produces nine full occlusions of any patch, if the patch is aligned to our stride grid. This mirrors what we have done with a stride of one. To account for patches not aligned to our stride grid, we increase our occlude by one stride less one pixel. Thus our occlude area is $(p + 3s - 1) \times (p + 3s - 1)$ pixels, for $s > 1$.

As an example, for CIFAR-10, we evaluate against a 5×5 attack patch, covering 2.4% of the image. For that, we occlude a 7×7 area, covering 4.8% of the image. With a stride of one, our prediction grid is 30×30.

If CIFAR-10 had ten times the resolution, 320×320, then the comparable sized attack patch would be 50×50 pixels, the same 2.4% of the image. For a stride of ten, our occlude area would be $(50+3\times10-1)\times(50+3\times10-1) = 79\times79$, or 6.1% of the image, more than before. Our prediction grid would be the same 30×30 size. However, we would be making predictions with more of the image occluded.

If occluding a larger percentage of the image was an issue, a 40×40 patch would allow a 69×69 occlude area. The predictions for the grid would thus have 4.7% of the image occluded, similar to before, with an expectation of comparable accuracy.

6 Experiments

We evaluate the effectiveness of our defense by measuring the clean accuracy (the images that when unmodified are classified correctly by class and as benign) and the certified accuracy (the images that when unmodified are classified correctly by class and as benign and where any attack – targeted or un-targeted – will either not change the classification or will be detected).

Data and Models We evaluate our defense on standard convolutional architectures, trained with data augmentation and random 90/10 train/validation splits. For CIFAR-10, we use SimpNet's 600K parameter version [HRF+18] trained for 700 epochs, though we do not yet reproduce all details of their training; for Fashion MNIST, a VGG-16 model [SZ14] trained for 50 epochs; for MNIST, the Deotte model [Deo18], with 40% dropout and batch normalization and 45 epochs. These serve as an inner model in our architecture.

Method We measure the clean and the certified accuracy on the 5000 or 6000 validation images. We perform multiple trials, using a different random 90/10 train/validation split for each trial. For each dataset, we perform $n = 4$ trials. The standard deviation is relatively low (for clean and certified accuracy they are CIFAR-10: 0.2 – 0.8% 0.5 – 1.1%, Fashion MNIST: 0.2–0.4% 0.2–0.6%, MNIST: 0.0 – 0.1% 0.1–0.5%). We report results for different points in the tradeoff between clean and certified accuracy, and we compare with recent related work using Interval Bounds Propagation (IBP) [CNA+20].

Results Our results, table 1, show that our defense achieves relatively high clean and certified accuracy and outperforms the previous state of the art.

For CIFAR-10, we achieve a clean accuracy of 92.4%, and 43.8% of images can be certified accurate (no matter where a sticker is placed, the resulting image will either be classified correctly or the attack will be detected) for 5×5 stickers. Our clean accuracy is 1.6% below that reported in the literature for the architecture we defend. It is only 0.1% below the accuracy we achieve with that architecture when evaluated on non-occluded images.

Table 1: The clean accuracy and certified accuracy of our defense (MR) vs. the previous state of the art (IBP) on all three datasets, for a 5×5 adversarial patch. We report the false positive rate of our defense in the third column; it is also included in the clean and certified accuracy. We report the literature reported accuracy of our inner model architectures in the fourth column. We report the accuracy our inner model achieves on non-occluded clean images in the fifth column.

			Accuracy			
Dataset	Defense	F.P.	Lit.	Inner	Clean	Cert.
CIFAR-10	IBP [CNA+20]				47.8%	30.3%
	MR (Our)	19.9%	94.0%	92.5%	78.8%	77.6%
		3.3%			90.6%	62.1%
		0.2%			**92.4**%	**43.8**%
Fashion	MR	12.9%		93.8%	85.4%	84.3%
		1.4%			93.0%	69.4%
		0.1%			93.9%	42.0%
MNIST	IBP [CNA+20]				92.9%	62.0%
	MR	4.8%	99.6%	99.6%	95.1%	94.9%
		0.7%			99.0%	75.8%
		0.2%			**99.4**%	**64.2**%

This is significantly better than recent work by Chiang et al. [CNA+20], which achieves clean accuracy of 47.8% and certified accuracy of 30.3% for CIFAR-10 against 5×5 stickers.

For MNIST, we achieve a clean accuracy of 99.4%, and 64.2% of images can be certified accurate for 5×5 stickers. This is again significantly better than recent work [CNA+20]: the error rate on clean images is more than an order of magnitude lower, and the certified accuracy is slightly higher.

Our measurement of certified accuracy is based on conservative assumptions. We suspect that many images that we cannot certify accurate are in fact secure against attack, even though we cannot prove it. Thus, the number certified accurate represents a conservative lower bound on the true robustness of our scheme.

Discussion Our experiments show that by choosing a high τ, we can achieve clean accuracy that is very close to the accuracy of our inner model on non-occluded images. With a lower τ we can achieve a higher certified accuracy at the cost of a lower clean accuracy.

For CIFAR-10, the architecture we used is reported to have an accuracy of 94.0% when trained appropriately. We did not replicate all aspects of the authors'

training procedure and achieved only 92.5%. Once we replicate their full training procedure, we expect our CIFAR-10 results would also improve.

We did an ablation study where we omitted the occlude training, and found that the occlude training is essential: Without it, the defense is extremely ineffective.

7 Limitations

Multiple patches would not be easy to handle with this approach, though they may also draw more attention to the attack. The simple extension would be all combinations of multiple occlude areas. For two patches, this would mean two occlude areas and a 4D prediction grid. That would be prohibitive in compute cost, and the multiple occludes would likely degrade accuracy.

Two patches might be present because the image is actually a binocular image. This would be straightforward to handle if the image came from a true physical scene and the parallax shift was not much. Widening the occlude area slightly would cover the two views of the same physical adversarial patch object.

The evaluation time cost of our defense is the size of the prediction grid, as for each occluded prediction, we predict on a new occluded image. It is possible lower layer convolutional results could be reused, but there would be a complexity cost, and we have not investigated this. For CIFAR-10 with a 5×5 patch, this is 900 times the evaluation cost of the original model. For a 320×320 pixel image, 50×50 patch, and stride 10, this is also 900 times the cost. We have not found any real difference in the time to train an occlude trained model than a normally trained one.

Our certified accuracy depends on the occluded accuracy of the architecture we defend. We have not examined datasets with lower top-1 accuracy, such as the 1000 class ImageNet. The more occluded predictions that are different, the more voting areas will not vote unanimously, causing the image to be vulnerable to attack.

Our defense is only effective against patches of irregular or unknown shape if they are bounded by the shape(s) we expect, of which one $n \times n$ shape is the most practical.

8 Related Work

In earlier work, Hayes proposes a defense against sticker attacks using inpainting of a suspected sticker region to remove the sticker from the image [Hay18]. This is similar to our defense. However, Hayes uses a heuristic to identify the region to inpaint (based on unusually dense regions within the saliency map), so any attack that fools the heuristic could defeat their defense. One could use inpainting in our scheme instead of occlusion, and it is possible this might improve accuracy, though our work can be viewed as showing that simple occlusion suffices to get strong results. Naseer et al. propose a defense against sticker attacks by smoothing

high-frequency image details to remove the sticker [NKP18]. They limit accuracy loss by using windows that overlap by a third, but their windows are smaller than the attack patch. Chiang et al. broke both of these defenses [CNA+20], so neither is effective against adaptive attacks; in contrast, we guarantee security against adaptive attack.

Wu et al. defend against adversarial patches with adversarial training [WTV20]. The primary advantage of our approach is that it provides certified security.

Chiang et al. study certified security against patch attacks using interval bounds propagation [CNA+20]. As discussed above, our defense achieves significantly better certified accuracy on both MNIST and CIFAR than their scheme. They also examine how their defense generalizes to other shapes of stickers and how to achieve security against L_0-bounded attacks, topics that we have not examined.

Zhang et al. limit the effect of a patch by clipping logits in a bag of features classifier and provide certified results [**?**]. Comparing our results with theirs is difficult as they use the higher resolution ImageNet dataset. They have higher robustness to attack but a larger cost to clean accuracy.

9 Conclusion

We propose the minority reports defense, a network architecture designed specially to be robust against patch attacks. We show experimentally that it is successful at defending against these attacks for a significant fraction of images.

Acknowledgments This work was supported by generous gifts from Google and Futurewei, by the Hewlett Foundation through the Center for Long-term Cybersecurity, and by Intel through the ISTC for Secure Computing.

References

BMR+17. Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. Adversarial patch, 2017, arXiv:1712.09665. 1, 4

CNA+20. Ping-yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Chris Studor, and Tom Goldstein. Certified defenses for adversarial patches. In *ICLR*, 2020. 6, 14, 15, 17

CW17. Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *Security and Privacy*, 2017, arXiv:1608.04644 [cs.CR]. 1

Deo18. Chris Deotte. How to choose CNN Architecture MNIST, 2018. https://www.kaggle.com/cdeotte/how-to-choose-cnn-architecture-mnist. 14

DT17. Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. 2017, arXiv:1708.04552 [cs.CV]. 6

EEF+17. Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models, 2017, arXiv:1707.08945. 1

GSS15. I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *ICLR*, 2015, arXiv:1412.6572 [stat.ML]. 1

Hay18. Jamie Hayes. On visible adversarial perturbations & digital watermarking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 16

HJN+11. Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I. P. Rubinstein, and J. D. Tygar. Adversarial machine learning, 2011. 1

HRF+18. Seyyed Hossein HasanPour, Mohammad Rouhani, Mohsen Fayyaz, Mohammad Sabokrou, and Ehsan Adeli. Towards principled design of deep convolutional networks: Introducing simpnet. *CoRR*, abs/1802.06205, 2018, 1802.06205. 14

KH09. Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 2

KZG18. Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. *CoRR*, abs/1801.02608, 2018, 1801.02608. 1

LBBH98. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998. 2

NKP18. Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. *CoRR*, abs/1807.01216, 2018, 1807.01216. 17

SZ14. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. arxiv:1409.1556. 14

SZS+14. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *ICLR*, 2014, arXiv:1312.6199 [cs.CV]. 1

TRG19. Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. 2019, arXiv:1904.08653. 1

USS+17. Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant CNNs. 2017, arXiv:1708.06500 [cs.CV]. 7

WTV20. Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. In *ICLR*, 2020. 17

XRV17. Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017, 1708.07747. 2

XZL+19. Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial T-shirt! Evading Person Detectors in A Physical World, 2019, arXiv1910.11099. 1

A Effects of Occlude Training

Our defense requires the inner model to handle occluded images well. To assess the effect of this requirement, we trained models with and without occlusions for all three inner-model architectures.

Training on occluded images appears to have only a small change on the accuracy of the inner model on non-occluded images, see table 2. The change is, at worst, the standard deviation of our measurements. Note from table 1 that

Table 2: The effect of training on occluded images, on the inner model's accuracy on non-occluded images. We show the difference (last column) and the standard deviation ($n = 4$).

	Type of training images		
Dataset	Non-occluded	Occluded	Δ
CIFAR-10	$92.5 \pm 0.3\%$	$92.5 \pm 0.2\%$	-0.0%
Fashion	$94.1 \pm 0.4\%$	$93.8 \pm 0.3\%$	-0.3%
MNIST	$99.58 \pm 0.08\%$	$99.63 \pm 0.33\%$	$+0.05\%$

the clean accuracy of our defense might have either a small or no drop from the accuracy of our inner-model.

Note that this does not measure the accuracy of our defense as a whole. Our defense feeds the inner model occluded images at test time, and accuracy on occluded images is slightly lower than on non-occluded images.

B Defense Details

The inner models are standard convolutional architectures modified to handle partially occluded data by the use of sparse convolutional layers that we created. The inner model returns a normal logit prediction for the dataset classes.

C Model Details

MNIST We used the Deotte model with layer descriptions ([32C3-32C3-32C5S2] - [64C3-64C3-64C5S2] - 128).