# Robust Defenses Against Physically Realizable Attacks on Image Classification

Sadam Hussain Ganie
*Department of Computer Science & Engineering*
*National Institute of Technology, Warangal - 506004, TS, India*
*sh23csm2r20@student.nitw.ac.in*

Dr. P. Radha Krishna
*Department of Computer Science & Engineering*
*National Institute of Technology, Warangal - 506004, TS, India*
*prkrishna@nitw.ac.in*

***Abstract*— The issue of safeguarding deep neural network methodologies for image classification against real-world physical attacks has been explored in [1]. Initially, the authors showcased the limited effectiveness of two prominent methods for building resilient models, namely adversarial training with projected gradient descent (PGD) attacks and randomized smoothing. Subsequently, they introduced a novel conceptual adversarial model known as rectangular occlusion attacks, where a malicious actor inserts a small, deliberately crafted rectangle into an image. They devised two methods to efficiently compute the resulting adversarial instances. Ultimately, they illustrated that employing adversarial training with their newly introduced attack enhances the robustness of image classification models against physically realizable attacks, presenting a pioneering generic defense mechanism. Our study focuses on optimizing the algorithm for placing rectangles on images by omitting calculations for irrelevant backgrounds and implementing gradient caching to prevent redundant computations. Additionally, we investigate the efficacy of incorporating noise during training, which demonstrates improved accuracy outcomes.**

***Keywords*— Rectangular occlusion attack - Projected gradient descent – CNN - Eyeglass frame - Stop sign sticker.***

## I. INTRODUCTION

Deep neural networks have become the de facto method in a number of fields, including speech recognition, computer vision, and natural language processing, due to their unmatched performance. Nevertheless, its effectiveness is undermined by deep neural networks' susceptibility to adversarial attacks, in which the network can be tricked by minute changes to image pixels. A lot of research has been done on ways to protect against these attacks; techniques range from detecting adversarial inputs to making neural network models more resistant. Demonstrations of physical attacks, in which adversarial perturbations are applied directly to real-world things, like stop sign stickers or eyeglass frames, are particularly concerning since they cause neural network classifiers to misclassify the objects. Effective strategies to fend against physical attacks are still difficult, despite significant attempts to guard against digital adversarial attacks.

There are two noteworthy contributions from the writers in [1]. First of all, it tests traditional methods of robust machine learning empirically against attacks that can be physically executed, namely the sticker attack on stop signs and the eyeglass frame attack on facial recognition. Examining how well adversarial training and randomized smoothing methods perform against these attacks shows how ineffective they are in this situation. Second, a unique abstract assault model that closely approaches frequent physically realizable attacks is introduced by the study: the rectangle occlusion attack. Under this model, a picture with adversarial selected content and location is covered by a rectangle sticker. Adversarial training yields neural network models resistant to these attacks, and several techniques are created to calculate such adversarial occlusions. Comparing the suggested methodology to existing methods, experimental results show that it is much more robust against physical attacks on deep neural networks.

Although the rectangular occlusion attack has shown some effectiveness in improving model robustness, its practical applicability is limited. In real-life scenarios, adversaries are not constrained to using rectangle-shaped patches for attacks; they can employ arbitrary shapes to perturb the input data. This flexibility allows attackers to exploit vulnerabilities in the model's decision boundary more effectively, as they can craft perturbations that are tailored to evade detection and mislead the classifier. Therefore, while the rectangular occlusion attack provides valuable insights into defense strategies, its narrow focus on a specific type of perturbation overlooks the broader spectrum of potential threats in real-world settings.

Robust learning techniques [2] have demonstrated potential because of their ability to withstand hostile attacks, despite the existence of numerous protection strategies for deep learning in vision applications. The most realistically effective technique is still adversarial training, especially when defending against attacks with $l_\infty$-norm perturbation constraints. Randomized smoothing [2] approaches have also been introduced recently as a powerful way to provide robustness, particularly against $l_2$-norm attacks. Furthermore, techniques for identifying adversarial cases have been investigated; nevertheless, their efficacy varies. The research is in line with current initiatives to describe how resilient neural networks are to physically plausible perturbations including translations, rotations, blurring, and contrast changes.

In real-life scenarios, particularly in applications like face recognition, the need for robust security measures is paramount. Achieving high accuracy in face detection, often exceeding 99%, is crucial for ensuring the reliability of such systems. However, attackers continually seek ways to manipulate these systems for malicious purposes. One significant challenge in this domain is face presentation attack detection, which aims to discern between genuine facial presentations and spoof attempts, such as the use of printed images or masks. Despite numerous efforts to develop effective face presentation attack detection methods, as documented in studies such as [26] and [27], the task remains highly challenging and largely unsolved. Researchers have explored various approaches to extract discriminative

features and enhance the capability of these systems to differentiate between real faces and presentation attacks. Nonetheless, the sophistication and diversity of presentation attack methods continue to pose significant hurdles to achieving robust and reliable face recognition systems in real-world environments.

In this study, we introduce three novel contributions. First, we optimize the rectangular attack algorithm by skipping loss calculations when the background is deemed irrelevant. Second, we employ gradient caching to streamline gradient calculations. Third, we enhance the model by introducing transformers and Gaussian noise, aiming to improve learning and prediction capabilities.

## II. Motivation

The existing literature has extensively addressed defending deep neural networks against adversarial examples in feature spaces like $l_2$ and $l_\infty$ norms. However, there remains a critical gap in effectively defending against physically realizable attacks, which pose significant concerns in real-world scenarios. When attackers introduce substantial noise into images, current models often misclassify them. Even advancements like the rectangular occlusion attack, which have shown improved accuracy, falter when faced with larger occlusion sizes, leading to a notable decrease in accuracy. Hence, there is a pressing need for novel techniques capable of mitigating these attacks and enhancing prediction accuracy, even in the presence of larger attack sizes.

Furthermore, in domains such as face recognition, where the integrity and security of systems are paramount, the stakes are particularly high. The demand for robust security measures, especially in critical applications like identity verification and access control, underscores the urgency of addressing vulnerabilities to physical attacks. Face presentation attack detection, in particular, remains a daunting challenge despite concerted research efforts. As attackers continually evolve their tactics, ranging from printed images to sophisticated masks, the need for advanced defense mechanisms becomes increasingly pressing.

Authors in [1], [2] have introduced noise into images and checked the accuracy. But due to the wide range of possible image attacks, including but not limited to those explored in these studies, it's imperative to develop techniques that can bolster accuracy even when faced with larger occlusion sizes in attacks, such as those observed in physically realizable scenarios. This necessitates a holistic approach to defense, encompassing not only traditional adversarial examples but also considering the myriad ways adversaries can exploit vulnerabilities in real-world settings.

## III. Background and Literature Review

Adversarial examples are subtle modifications made to input images, often imperceptible to humans, yet capable of systematically misclassifying state-of-the-art neural networks. These attacks, extensively studied in the literature, involve optimizing an objective function to find perturbations that maximize misclassification while adhering to a specified constraint, typically based on a norm such as $l_2$ or $l_\infty$. Among these attacks, Carlini & Wagner (2017b) [3] and Madry et al. (2018) [4]'s projected gradient descent (PGD) attack are noteworthy.

While existing research primarily focuses on attacks that directly alter digital images, we shift our attention to a class of physical attacks that modify real-world objects to deceive neural networks. These attacks share three key attributes: they can be implemented in physical space, they maintain low suspicion by altering only a small portion of the object, and they induce misclassification by advanced deep neural networks. Although our primary concern is defense, we analyze the digital representations of these physical attacks, disregarding practical implementation details like viewpoint robustness and printability. Thus, we term these attacks "physically realizable attacks" to emphasize their feasibility in practice, although their digital simulations represent a stronger adversarial model.

We study three methods that are physically feasible: the adversarial patch assault that uses stickers with adversarial noise, the stop sign attack that uses adversarial stickers, and the eyeglass frame attack on facial recognition. These attacks try to change real-world items so that neural networks misclassify them.

We tackle the problem of adversarial resilience in deep learning by emphasizing defense tactics that are based on principles and have not been rendered inoperable by advanced attacks. These techniques can be divided into two primary groups: randomized smoothing and robust learning. The goal of robust learning is to minimize a robust loss function, which is frequently estimated by adversarial training methods as PGD. A curriculum-based variant is also taken into consideration. In contrast, randomized smoothing incorporates noise into inputs at both the training and prediction stages, resulting in a smoothed classifier that takes into consideration the generated decision distribution. Both strategies present viable paths toward proving strong classification against adversarial attacks.

Following are a few examples of adversarial attacks on images.


(a) (b)

Figure 1: (a) An illustration of an attack using eyeglass frames. Original facial input image on the left. Middle: The image has been altered by adding glasses to it. Right: a picture of the anticipated person with the adversarial input in the center picture. (b) An illustration of the attack on stop signs. Original stop sign input image on the left. adversarial mask in the middle. Right: stop sign image with adversarial stickers, classified as a speed limit sign.

Figure 2: Examples of the ROA attack on face recognition, using a rectangle of size 100 x 50. (a) Left: the original A. J. Buckley's image. Middle: modified input image. Right: an image of the predicted individual who is Aaron Tveit. (b) Left: the original Abigail Spencer's image. Middle: modified

input image. Right: an image of the predicted individual who is Aaron Yoo with the adversarial input in the middle image.

Adversarial examples, as conceptualized by [5], involve starting with a classifier $f: R^n \rightarrow \{1, .., k\}$ and an original sample $X \in R$ (often a benign sample from the training set). The goal is to identify the smallest possible modifications to x, denoted as $\bar{x}$, that results in a change in the classification output by f: f(x) $\neq$ f($\bar{x}$). In simpler terms, adversarial examples are crafted by tweaking the original sample just enough to fool the classifier into making a different prediction.

The adversarial training, initially introduced by Goodfellow et al. [6], has been an effective defense strategy. This approach introduces adversarial images into the training phase to strengthen the model against the adversarial attacks. However, its efficacy is limited to certain kinds of attacks only whereas it is not effective against other attacks.

Generative models have attracted attention as potential tools for adversarial purification due to their ability to generate or transform data [7, 8, 9]. Among these, diffusion models have emerged as promising candidates. Recent research, such as that by Nie et al. [10] and Carlini et al. [11], has explored the use of score-based and diffusion models for purifying adversarial samples.

Current adversarial purification techniques have primarily concentrated on images, overlooking the potential of diffusion models in multi-modal tasks like text-to-image generation [12].

Image diffusion refers to the process of smoothing input images, achieved either through constant-rate smoothing (linear diffusion) [13], where smoothness is uniformly applied across the image, or through nonlinear methods that preserve essential features like edges [14], [15]. In computer vision, linear diffusion, a classic partial differential equation method, is widely studied, representing an evolution process where images are uniformly smoothed in all directions at a consistent pace. However, such diffusion tends to diminish finer structural details in the image. Modern image diffusion techniques encompass both linear and nonlinear models. Gaussian smoothing stands out as a popular method among linear schemes [16]. Among nonlinear approaches, Perona-Malik diffusion, also known as anisotropic diffusion, is prevalent in image processing [14]. Additional nonlinear methods include continuous, semi-discrete, and discrete diffusion filtering schemes [16]. Hybrid diffusion, modified Perona-Malik models, and others are also part of the diverse landscape of image diffusion models [17], [18], [19].

Theoretical sensitivity analyses, as discussed by researchers in [20, 21, 22, 23], delve into the effects of parameter perturbations. Shu and Zhu [20] introduce an influence measure inspired by information geometry to gauge how various perturbations to input signals and network parameters impact DNN classifiers. Xiang et al. [23] devise an iterative method to calculate the sensitivity of DNN layers step by step, defining sensitivity as the expected absolute output variation resulting from weight perturbations across all possible inputs. Tsai et al. [21] explore the robustness of pairwise class margin functions against weight perturbations, while Weng et al. [22] determine a certified robustness threshold for weight perturbations, ensuring that a neural network maintains accurate outputs within this range. Furthermore, they establish a valuable link between this certification and the challenge of weight quantization.

## IV. PROPOSED METHOD

In Section 3, it became evident that traditional models aimed at bolstering deep learning against attacks struggle significantly when confronted with physically realizable threats. This suggests that conventional attack models, which typically involve lp-bounded perturbations to input images, are inadequate for addressing the primary physical threats encountered in practice. It highlights the question of whether it is possible to create a complete strategy that ensures robustness against physical attacks given the variety of potential attacks. Addition of hostile occlusions to portions of the input is a common feature of physical attacks. These attacks can take on different forms or locations, but they always need to avoid suspicion and reduce the size of the adversarial occlusion. As a result, we present an abstract model of occlusion assaults in its simplest form and investigate techniques for their computation as well as methods to make classifiers more resilient to them.

The following simple abstract model of adversarial occlusions in input images is proposed by the authors in [1]. Anywhere in the image, the attacker inserts a fixed-dimension rectangle and has the opportunity to add $l_\infty$ noise inside it up to a predefined upper bound $\varepsilon$ (e.g., $\varepsilon = 255$, which allows arbitrary adversarial noise to be added). While this model mirrors common physical limits, it also shares several properties with l0 assaults. Specifically, it imposes a contiguity constraint through the rectangle. Despite being abstract, the model captures important details shared by many physical attacks, like stickers applied to a specific area of the target. Termed a rectangular occlusion attack (ROA), this model is noteworthy for being untargeted: by focusing on untargeted attacks, precise knowledge of the attacker's objectives is unnecessary, aligning with the overarching goal of defending against physical attacks regardless of their specific targets.

In Figure 3 and Figure 4, we illustrate examples of rectangular occlusion attacks applied to images from the MNIST and CIFAR10 datasets, respectively. In the first case, a rectangle of size 5x5 pixels is placed on the image whereas in second case, a rectangle of size 10x10 is placed on the image in such a way that the loss incurred by the classifier is maximized.

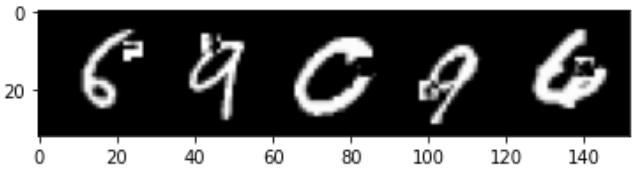


Figure 3: ROA attack of size 5x5 on MNIST dataset image.

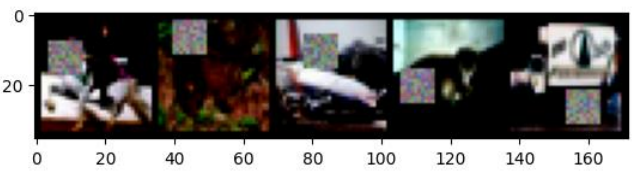


Figure 4: ROA attack of size 10x10 on CIFAR10 dataset image.

There are two primary processes in computing ROA attacks:
1. Deciding where in the image the rectangle should be placed.
2. Producing localized adversarial perturbations that are fine-grained.
First, a comprehensive search can be used to determine every possible location for the rectangle's upper left corner. Next, adversarial noise inside the rectangle can be calculated for each position using Projected Gradient Descent (PGD), and the worst-case attack that maximizes loss on the final image can be chosen. However, because PGD needs to be done inside the rectangle for each possible place, this method may be computationally demanding.
In [1], their method decouples these tasks in order to address this difficulty. To find a site that maximizes loss, they first perform a thorough search with a gray rectangle. Once this is fixed, PGD is only applied inside the rectangle.
Nevertheless, a major drawback of the exhaustive search approach is that it needs to compute the loss function for each potential position, which means that full forward propagation is required each time. As a result, the search is still going very slowly. The authors in [1] find candidate places by using the gradient of the input image to increase efficiency. To be more precise, [1] narrow down the exhaustive search to a subset of C sites that have the maximum gradient magnitude for the sticker. The number of loss function computations is greatly decreased when C is found to be minimal in relation to the total number of pixels in the image.
After calculating the ROA assault, [1] implement the conventional adversarial training strategy for defense. The resulting classifiers, dubbed Defense against Occlusion assaults (DOA) by the authors in [1], are resistant against our abstract adversarial occlusion assaults. [1] suggest using these classifiers in place of traditional robust machine learning (ML) to protect against physical assaults. This defense against ROA is successful for our goals, as shown.
We propose an innovative strategy to streamline the process of placing a rectangle on an image, leveraging both exhaustive search and gradient-based methods. These traditional approaches are known to be computationally demanding and time-consuming, often requiring substantial resources to execute.
Our proposed optimization revolves around identifying and capitalizing on regions of the image where the placement of the rectangle has little to no impact on the subsequent loss calculation. Specifically, we introduce the concept of "irrelevant regions", which pertain to areas of the image that do not significantly contribute to the classification task. For instance, in datasets like MNIST or CIFAR10, where images may contain uniform backgrounds (e.g., entirely black or white regions), these areas can be deemed irrelevant for the classification task.
By implementing a criterion that assesses the relevance of the region, we can selectively skip the loss calculation step when the rectangle is positioned in such areas. Our criterion dictates that if more than 50% of the rectangle background corresponds to an irrelevant region, the loss calculation is bypassed. This strategy effectively reduces the number of computations required, resulting in notable time savings during the process.
Moreover, we introduce a caching mechanism to store and reuse gradients computed during the gradient-based method. By storing these gradients, we can avoid redundant calculations when encountering similar regions in subsequent iterations. This approach minimizes computational redundancy and further accelerates the process of placing the rectangle on the image.
Our novel techniques aim to optimize the efficiency of both exhaustive search and gradient-based methods, facilitating faster and more resource-efficient computation. By strategically skipping unnecessary calculations and leveraging cached gradients, we significantly reduce the time required for the rectangle placement process, thereby enhancing the overall efficiency of the adversarial defense mechanism.
There are two primary steps in the algorithm in [1] for calculating Rectangular Occlusion Attacks (ROA):
1. Positioning the Rectangle: This phase involves moving the rectangle's upper left corner point across the image in a number of feasible locations. The objective of [1] is to determine the best location for a gray rectangle (RGB value = [127.5,127.5,127.5]) in order to optimize the adversarial attack's loss function.
2. Introducing Perturbations: [1] generates strong $l_\infty$ noise inside the rectangle at a given place after determining the rectangle's ideal location.
Algorithm 1 presents the whole algorithm for locating the ROA point. It utilizes a comprehensive search strategy throughout the image's pixel region. This algorithm uses multiple parameters. To begin with, [1] assumes that the images have $N^2$ dimensions and are square. Second, in order to speed up the location computation procedure, [1] introduces a stride parameter (S). Through the use of a stride parameter, [1] essentially skips S pixels every time by only taking into account every other $S^{th}$ pixel during the search. Select the stride value S = 5 for face recognition tasks and S = 2 for traffic sign classification in order to conduct ROA attacks [1]. Adjusting the stride parameter allows us to balance computational efficiency with the granularity of the search process, optimizing the performance of the ROA algorithm for different classification tasks.

**Algorithm 1** Computation of ROA position using exhaustive search.

**Input:**
Data: $X_i, y_i$; Test data shape: $N \times N$; Target model parameters: $\theta$; Stride: $S$ ;
**Output:**
ROA Position: $(j', k')$
1.**function** ExhaustiveSearching($Model, X_i, y_i, N, S$)
2. **for** j **in** range($N/S$) **do**:
3. **for** k **in** range($N/S$) **do**:
4. Generate the adversarial $X_i^{adv}$ image by:
5. place a grey rectangle onto the image with top-left corner at $(j \times S, k \times S)$;
6. **if** L($X_i^{adv}, y_i, \theta$) is higher than previous loss:
7. **Update** $(j', k') = (j, k)$
8. **end for**
9. **end for**
10. **return** $(j', k')$

Figure 5: Calculation of ROA placement using exhaustive search.

Finding the best place for the Rectangular Occlusion Attack (ROA) still necessitates a significant number of loss function evaluations, which can be computationally costly because each evaluation requires a full forward pass through the deep neural network. This is even with the introduction of the tunable stride parameter. In an effort to lessen the computing load, [1] investigate the potential of using the gradient magnitude of the loss as a gauge for how manipulable particular regions are.

To be more precise, [1] calculates the gradient ∇L and looks at the gradient values ($\nabla L_i$) for each pixel in the picture. [1] is able to evaluate the sensitivity of any possible ROA position by summing the squared gradient values over pixels inside the rectangular region. By using this procedure, [1] is able to prioritize regions that are more manipulatable by determining the top C possible places for the rectangle. In order to determine the ideal site for the ROA, [1] then assess the loss function for every potential location. Algorithm 2 presents a detailed algorithm describing this methodology.

Once the optimal location for the rectangle is determined, the next step involves introducing adversarial noise within the identified region. To achieve this, [1] employ the $l_\infty$ version of the Projected Gradient Descent (PGD) attack, confining perturbations strictly within the rectangle. In order to produce adversarial noise inside the rectangle, the authors in [1] run a predetermined number of PGD iterations ({7, 20, 30, 50}), each with a matching learning rate α {32, 16, 8, 4} that is specifically designed to maximize the attack process. Through this iterative process, adversarial perturbations can be systematically introduced within the designated region by [1], improving the overall robustness of the final attack method.

**Algorithm 2** Computation of ROA position using gradient-based search.

**Input:**
Data: $X_i, y_i$; Test data shape: $N \times N$; Target Model: $\theta$; Stride: $S$;
Number of Potential Candidates: $C$;
**Output:**
Best Sticker Position: $(j', k')$

**function** GradientBasedSearch($X_i, y_i, N, S, C, \theta$)
  Calculate the gradient $\nabla L$ of Loss($X_i, y_i, \theta$) w.r.t. $X_i$
  $\mathbb{J}, \mathbb{K}$ = HelperSearching($\nabla L, N, S, C$)
  **for** $j, k$ **in** $\mathbb{J}, \mathbb{K}$ **do**:
    Generate the adversarial $X_i^{adv}$ image by:
    put the sticker on the image where top-left corner at $(j \times S, k \times S)$;
    **if** Loss($X_i^{adv}, y_i, \theta$) is higher than previous loss:
      **Update** $(j', k') = (j, k)$
  **end for**
  **return** $(j', k')$

**function** HelperSearching($\nabla L, N, S, C$)
  **for** j **in** range($N/S$) **do**:
    **for** k **in** range($N/S$) **do**:
      Calculate the Sensitivity value $L = \sum_{i \in \text{rectangle}} (\nabla L_i)^2$ where top-left corner at $(j \times S, k \times S)$;
      **if** the Sensitivity value $L$ is in top $C$ of previous values:
        Put $(j, k)$ in $\mathbb{J}, \mathbb{K}$ and discard $(j_s, k_s)$ with lowest $L$
    **end for**
  **end for**
  **return** $\mathbb{J}, \mathbb{K}$

Figure 6: Calculation of ROA placement using gradient-based search.

A visual representation of the comparison between gradient-based search and exhaustive search for calculating ROA may be found in Figure 7.

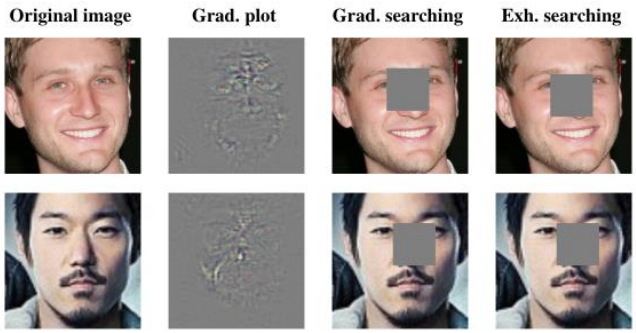


Figure 7: Examples of different search techniques. 1) The original input image 2) The plot of the input gradient 3) The face whose location was determined by ROA through gradient-based search 4) The face whose location was determined by exhaustive search.

Figure 8 illustrates our proposed algorithm for optimizing the computation process by skipping the loss calculation when the background of the image is either fully black or fully white. The algorithm operates by assessing the background surrounding the region where the rectangle is intended to be placed.

To determine the background, we consider a region with dimensions equal to half the width and half the height of the rectangle we aim to position on the image. By analyzing this background region, we can ascertain whether it predominantly consists of black or white pixels.

In the case of the MNIST dataset, which typically features digits on a white background, a background closer to white indicates that the surrounding area is predominantly white. Conversely, if the background is closer to black, it suggests that the surrounding region is predominantly black.

Based on this analysis, if the algorithm identifies that the background is indeed fully black or fully white, it omits the calculation of the loss function. This decision is made because in such scenarios, the presence of the rectangle is unlikely to significantly impact the classification outcome, as the background itself is uniform and does not provide meaningful information for classification. By skipping the loss calculation in these instances, the algorithm conserves computational resources and expedites the overall process of identifying the optimal placement for the rectangle.

**Algorithm** Skipping Calculation when Background is Close to White

**for** $i \leftarrow 0$ **to** $xtimes$ **do**
  **for** $j \leftarrow 0$ **to** $ytimes$ **do**
    $region \leftarrow X[:, :, yskip \times j : (yskip \times j + \text{height}//2), xskip \times i : (xskip \times i + \text{width}//2)]$
    $avg\_pixel\_value \leftarrow \text{torch.mean}(region)$
    **if** $avg\_pixel\_value > 0.9$ **then**
      $skip\_ctr \leftarrow skip\_ctr + 1$
      **continue**
        ▷ Skip further checking if background is close to white
    **end if**
  //Remaining code
  **end for**
**end for**

Figure 8: Algorithm 3; Skipping calculation when background is close to white.

Next, we introduce the gradient caching algorithm, depicted in Figure 9. This algorithm efficiently stores and retrieves precomputed gradients during the training process. It works by first computing the gradients and loss for a given input configuration. If the same input configuration is encountered again during training, the algorithm retrieves the cached gradients instead of recomputing them. This approach optimizes training time, conserves computational resources, ensures training stability, and scales effectively with larger datasets.

Figure 10 illustrates the architecture of the model used in [1]. This work uses a simple CNN model for adversarial training on the MNIST dataset [24]. This model operates by taking images as input, incorporating the rectangular occlusion technique described earlier, and then undergoes training. Following the training process, the model calculates its accuracy. This CNN model is characterized by its simplicity

and is specifically designed for image classification tasks on the MNIST dataset.

**Algorithm** Gradient Caching Algorithm

Initialize empty dictionary $cached\_gradients$
Initialize $gradient$ and $X1$ tensors with gradients enabled
Compute input key $(X.shape, y.shape, device)$
**if** input key not in $cached\_gradients$ **then**
    Compute loss and gradients using the model
    Update $cached\_gradients$
**else**
    Retrieve $cached\_gradients$
**end if**
**return** $cached\_gradients$

Figure 9: Algorithm 4; Gradient caching algorithm.

Using the MNIST dataset, the CNN model architecture for adversarial training comprises two convolutional layers, followed by max pooling, batch normalization, and ReLU activation. Eight output channels and a kernel size of three are present in the first convolutional layer, whereas 32 output channels and a kernel size of five are present in the second convolutional layer. Following each convolutional layer, the feature maps are down sampled using max pooling, batch normalization, and ReLU activation algorithms. After being flattened, the second max pooling operation's output is then sent via two fully linked layers with, respectively, 600 and 10 output features. In order to avoid overfitting, a dropout layer with a dropout probability of 0.5 is applied before the second fully connected layer. The model is trained and tested on two datasets viz. MNIST, & CIFAR10[25].

The model's performance tends to degrade as the size of the occluding rectangle increases. When the occlusion size is relatively small, such as 5x5 pixels or less, the model maintains higher accuracy. However, as the size of the occlusion surpasses this threshold, for instance, reaching 10x10 pixels as implemented in our proposed work, the model's accuracy starts to decline.

This decline in accuracy can be attributed to how the occlusion affects the features extracted by the model. Smaller occlusions may obscure fewer critical features in the input image, allowing the model to still make accurate predictions. However, larger occlusions cover more significant portions of the image, potentially hiding crucial features that the model relies on for classification. Consequently, the model's performance suffers, leading to decreased accuracy.

In our project, we experimented with training the model using larger occlusions of size 10x10 pixels. This was done to evaluate the model's robustness to larger occlusions and its ability to generalize to more challenging scenarios. However, the observed decrease in accuracy underscores the importance of carefully considering the size and impact of occlusions when training models for robustness against adversarial attacks.

To enhance the robustness of our model against adversarial attacks, we introduced several modifications to the CNN architecture mentioned above, as shown in figure 11. Specifically, we incorporated transformers into the existing CNN network and augmented the training process by adding Gaussian noise.

The inclusion of transformers introduces an attention mechanism into the model, allowing it to focus on relevant parts of the input data while suppressing irrelevant information. This enables the model to better capture intricate patterns and subtle features in the input images, thereby improving its overall performance and resilience against adversarial perturbations.

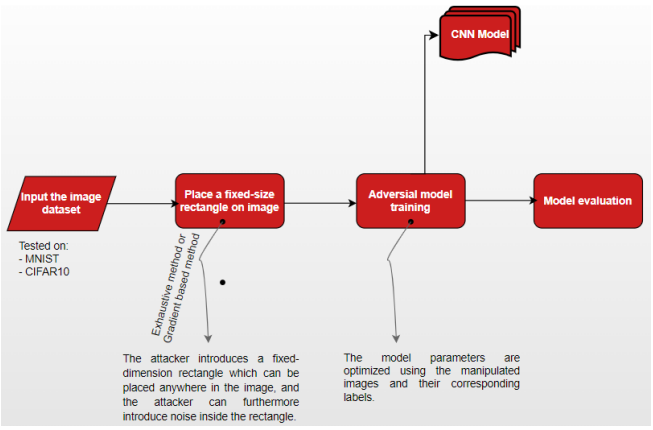


Figure 10: Architecture of the model used in [1].

Additionally, we introduced Gaussian noise during the training phase to augment the dataset. By injecting random noise into the input images, we simulate real-world variations and increase the model's robustness to noisy or distorted inputs. This regularization technique helps prevent overfitting and enhances the generalization ability of the model, making it more adept at handling unseen or adversarial crafted examples.

Overall, these modifications contribute to the development of a more robust and reliable model capable of effectively mitigating the impact of adversarial attacks on image classification tasks.

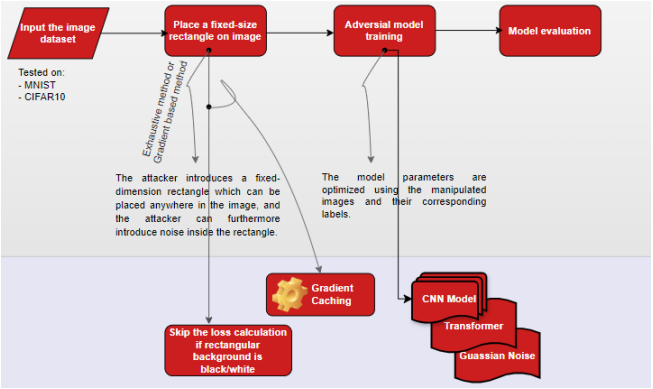


Figure 11: Proposed model architecture.

The main architecture of the proposed model which has CNN, Transformer, & Gaussian Noise comprises several components to enhance robustness and performance in image classification tasks.

### *A. GaussianNoise Module*

This module introduces Gaussian noise into the input data, aiding in regularizing the model during training to prevent overfitting and improve generalization. The level of noise is controlled by the parameter sigma.

### *B. ConvNet Module*

The core of the model is this module, which extracts useful characteristics from the input images using a sequence of

convolutional layers, activation functions, and pooling operations. Dropout layers are optionally added to further reduce overfitting. Both the dropout rates and the number of dropout layers are adjustable.

### C. Transformation Layer

This layer, though commented out in the current implementation, can be activated to incorporate transformer-based processing. Transformers are powerful models for sequence-to-sequence tasks and can capture long-range dependencies in data.

The architecture's forward pass involves passing the input through the convolutional layers, applying ReLU activation functions, and down sampling through max-pooling operations. Optionally, dropout is applied to the feature maps before flattening them for input to fully connected layers. Gaussian noise is injected again before passing the data through the linear layers, and finally, the output is passed through a log softmax function to obtain the class probabilities.

By incorporating Gaussian noise and optionally utilizing transformers, this architecture aims to enhance the model's robustness and performance, particularly in the face of adversarial attacks and noisy inputs. Adjustments to dropout rates, noise levels, and the inclusion of transformer layers provide flexibility in tailoring the model to specific tasks and datasets.

## V. Dataset Analysis

Following datasets have been used in the model evaluation. The analysis of the datasets is as follows.

### A. MNIST

A popular benchmark dataset for image classification problems is the MNIST dataset, which consists of grayscale pictures of handwritten numbers from 0 to 9. Here, we present an examination of the MNIST dataset and discuss how it relates to our work on training adversaries using rectangular occlusion techniques. The MNIST dataset is made up of 10,000 test images and 60,000 training images, each measuring 28 by 28 pixels. The ten-digit classifications are evenly distributed across these photos, creating a balanced dataset that can be used for both training and assessing machine learning models.

We preprocess the MNIST images by converting them to PyTorch tensors and normalizing pixel values to the range [0, 1] before training our models. To further enhance the generality and robustness of the model, we also add random modifications to the dataset, such as translation, scaling, and rotation.

We train our models on the MNIST dataset using various architectures, including convolutional neural networks (CNNs) and hybrid models combining CNNs with attention mechanisms (transformers). During training, we employ techniques such as data augmentation, dropout regularization, and adversarial training to enhance model performance and resilience to adversarial attacks.

We analyze the models' robustness against adversarial attacks, specifically rectangular occlusion attacks, by measuring their accuracy under varying attack intensities and sizes of occluded regions.

### B. CIFAR10

Another popular benchmark dataset for image classification tasks is the CIFAR10 dataset, which consists of color photos of ten different item classes, including cars, dogs, cats, and airplanes. Here, we present a study of the CIFAR10 dataset and discuss how it relates to our work on training adversaries using rectangular occlusion techniques.

The CIFAR10 dataset consists of 10,000 test images and 50,000 training images, each with three RGB color channels and a size of 32 by 32 pixels. The photos are dispersed equally among the ten item classes, much like MNIST, which makes it appropriate for training and assessing classification models.

We preprocess the CIFAR10 images in the same way as we did with the MNIST dataset, normalizing pixel values and transforming them to PyTorch tensors. To improve the diversity of training samples, we also apply data augmentation techniques including random rotations, random cropping, and random horizontal flips.

We use comparable architectures and training procedures as we did with MNIST to train our models on the CIFAR10 dataset. To attain the best results, we might use deeper CNN architectures and more advanced training methods because of the greater complexity and variety of CIFAR10 images.

The same measures we use for MNIST are also used to assess model performance on CIFAR10, including accuracy. We also evaluate the resilience of the model against adversarial assaults, namely rectangular occlusion attacks, in order to ascertain how well our suggested defense measures work.

Training deep neural network models on large datasets like CIFAR-10 can indeed be time-consuming due to the dataset's size and the complexity of the model architecture. To mitigate this challenge, we trained our CNN-Transformer-Gaussian Noise model on a subsampled version of the CIFAR-10 dataset, specifically utilizing only 10% of the dataset for training.

The results of our experiments indicate that while the accuracy achieved on CIFAR-10 may not match that attained on simpler datasets like MNIST, we have made notable progress compared to existing techniques proposed in [1]. Despite the reduced dataset size, our model demonstrates improved performance, showcasing its efficacy in handling the complexities inherent in real-world image classification tasks.

By leveraging the combined power of convolutional neural networks, transformer layers, and Gaussian noise regularization, our model exhibits enhanced robustness and generalization capabilities. Our results highlight the promising potential of our proposed approach in advancing the state-of-the-art in defending deep neural network methodologies against adversarial attacks and real-world challenges in image classification, even though more study and testing may be necessary to achieve higher accuracy levels on CIFAR-10.

## VI. Result Analysis

The results of our experiments, conducted on a system with the specified specifications, are summarized in Table 1. Here's an explanation of the results:

Number of Epochs: We trained our models for 5 epochs on both datasets.

Dataset Size: We trained the models on the entire MNIST dataset (100%) and a subset of the CIFAR10 dataset (10%).
Execution Time: For the gradient caching and background skip algorithm, our modified model showed a reduction in execution time compared to the base model. Specifically, for 1 epoch, it took 20 seconds less than the base model on 10% of MNIST dataset.

| | |
|---|---|
| CPU | Core i5, 10th Generation |
| RAM | 8 GB |
| Storage | 256 GB SSD |
| Graphics | 2 GB |
| Processors | 8 |

Table 1: System specifications

| Dataset | Model | Test Accuracy | Adversarial Accuracy | Attack Size |
|---|---|---|---|---|
| MNIST | [1]model | 81.80% | 41.26% | 10x10 |
| MNIST | Our proposed model | 89.45% | 72.80% | 10x10 |
| CIFAR10 | [1]model | 15.10% | 10.50% | 10x10 |
| CIFAR10 | Our proposed model | 16.80% | 12.00% | 10x10 |

Table 2: Results

### *A. MNIST Results*

Base Model: The base model presented in [1] achieved a test accuracy of **81.8%** on the MNIST dataset, but its adversarial accuracy dropped significantly to **41.26%** when subjected to 10x10 pixel attacks.
Our Model: In contrast, our modified model achieved a higher test accuracy of **89.45%** on MNIST and demonstrated improved robustness against adversarial attacks, with an adversarial accuracy of **72.8%** under the same attack size.

### *B. CIFAR10 Results*

Base Model: The base model achieved a lower test accuracy of **15.10%** on the CIFAR10 dataset, and its adversarial accuracy further decreased to **10.50%** when subjected to 10x10 pixel attacks.
Our Model: Our modified model exhibited a slightly higher test accuracy of **16.8%** on CIFAR10 and showed improved resilience against adversarial attacks, with an adversarial accuracy of **12%** under the same attack size.
Overall, the findings show how well our suggested model adjustments work, especially when applied to the MNIST dataset, to improve neural network test accuracy and robustness against adversarial attacks. Still, there's potential for development, particularly with more intricate datasets like CIFAR10. Future revisions may perform better as a result of further testing and optimization.

## VII. CONCLUSION

In this project, we investigated adversarial attacks, particularly focusing on physically realizable attacks that manipulate real-world objects to deceive neural networks. Our exploration encompassed various defence strategies, including robust learning and randomized smoothing, to address the challenge of adversarial robustness in deep learning.
We introduced the concept of rectangular occlusion attacks (ROA) and proposed defence mechanisms against them, including an abstract model for occlusion attacks and efficient algorithms for computing ROA positions. Leveraging both exhaustive search and gradient-based methods, we optimized the process of placing rectangles on images, enhancing the efficiency of adversarial defence mechanisms.
In addition to the advancements in adversarial defence strategies and model modifications, our project underscores the importance of robustness testing and evaluation methodologies in assessing the efficacy of defence mechanisms. By rigorously evaluating our proposed approaches on diverse datasets and under varying attack intensities, we provide valuable insights into the strengths and limitations of current defence techniques. These findings pave the way for future research directions aimed at developing more resilient and trustworthy deep learning systems capable of withstanding adversarial challenges in real-world applications. Through continued collaboration and innovation, we strive to foster a safer and more secure landscape for machine learning deployment, ultimately empowering the widespread adoption of AI technologies across various domains.
In addition, we suggested modifying the CNN design by adding transformers and Gaussian noise to strengthen the model's resistance to hostile attacks. Our tests using the MNIST and CIFAR10 datasets showed how well our suggested model adjustments worked to increase test accuracy and defend against adversarial attacks.

## VIII. ACKNOWLEDGMENT

All authors contributed to the study conception, research work and project implementation. All authors read and approved the final manuscript.

## IX. REFERENCES

[1] "Defending Against Physically Realizable Attacks on Image Classification," Tong Wu, Liang Tong, and Yevgeniy Vorobeychik, ICLR 2020.
[2] M. Lecuyer, S. Jana, D. Hsu, R. Geambasu, and V. Atlidakis. Adversarial examples with differential privacy are certified robust. In the 2019 IEEE Symposium on Security and Privacy.
[3] D. Wagner and N. Carlini. Adversarial instances are difficult to identify: eschewing eleven methods of detection. In International AI and Security Workshop, 2017a.
[4] Dimitris Tsipras, Adrian Vladu, Dimitris Madry, and Aleksandar Makelov. In the direction of adversarial attack-resistant deep learning models. In the 2018 International Conference on Learning Representations.
[5] C. Szegedy et al. (2013) arXiv:1312.6199, "Intriguing properties of neural networks."
[6] Christian Szegedy, Jonathan Shlens, and Ian J. Goodfellow, "Explaining and harnessing adversarial examples," in ICLR, 2015.

[7] Pouya Samangouei, Maya Kabkab, and Rama Chellappa, "Defense-gan: Use generative models to protect classifiers against adversarial attacks," in ICLR, 2018.
[8] "Online adversarial purification based on self-supervised learning," Changhao Shi, Chester Holtz, and Gal Mishne, in ICLR, 2020.
[9] "Adversarial purification with score-based generative models," by Jongmin Yoon, Sung Ju Hwang, and Juho Lee, in ICML, 2021.
[10] Diffusion models for adversarial purification, by Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar, in ICML, 2022.
[11] "(certified!!) adversarial robustness for free!" by Nicholas Carlini, Florian Tramer, Krishnamurthy Dj Dvijotham, Leslie Rice, Mingjie Sun, and J Zico Kolter, in ICLR, 2022.
[12] "High-resolution image synthesis ¨ with latent diffusion models," by Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer, in CVPR, 2022.
[13] Linear Diffusion, E. Erdem. Hacettepe University, Ankara, Turkey, February 2012.
[14] "Anisotropic diffusion," in Geometry-Driven Diffusion in Computer Vision, by P. Perona, T. Shiota, and J. Malik. Springer, Cham, Switzerland, 1994, pp. 73–92.
[15] "Nonlinear anisotropic filtering of MRI data," G. Gerig, O. Kubler, R. Kikinis, and F. A. Jolesz, IEEE Trans. Med. Imag., vol. 11, no. 2, pp. 221–232, Jun. 1992.
[16] J. Weickert, Teubner Stuttgart, 1998, Stuttgart, Germany, Anisotropic Diffusion in Image Processing, vol. 1.
[17] In 2012, Pattern Recognition published a paper titled "Reducing aliasing in images: A PDE-based diffusion revisited," written by D. Ziou and A. Horé.
[18] "Image denoising using modified Perona–Malik model based on directional Laplacian," by Y. Q. Wang, J. Guo, W. Chen, and W. Zhang, Signal Process., vol. 93, no. 9, pp. 2548–2558, Sep. 2013.
[19] "A hybrid model for image denoising combining modified isotropic diffusion model and modified Perona–Malik model," N. Wang, Y. Shang, Y. Chen, M. Yang, Q. Zhang, Y. Liu, and Z. Gui. IEEE Access, 6 (December 2018), 33568–33582.
[20] Hongtu Zhu and Hai Shu. Analysis of deep neural networks' sensitivity. 2019; ArXiv, abs/1901.07152.
[21] Pin-Yu Chen, Yu-Lin Tsai, Chia-Yi Hsu, and Chia-Mu Yu. giving formal form to 22 neural network adversarial robustness and generalization against weight perturbations. 34:19692–19704, Advances in Neural Information Processing Systems (NeurIPS), 2021.
[22] Pin-Yu Chen, Xue Lin, Sijia Liu, Pu Zhao, Tsui-Wei Weng, and Luca Daniel. In the direction of certified model resilience to weight fluctuations. 34(04):6356–6363, Proceedings of the AAAI Conference on Artificial Intelligence, April 2020.
[23] Yanjun Liu, Yuhu Niu, Xiaoqin Zeng, and Lin Xiang. investigation of convolution neural networks' sensitivity to weight perturbations. 7:93898–93908, IEEE Access, 2019.
[24] In IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141-142, Nov. 2012, L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," doi: 10.1109/MSP.2012.2211477.
[25] "Learning Multiple Layers of Features from Tiny Images," by Alex Krizhevsky, 2009.
[26] In November 2015, IEEE Trans. Inf. Forensics Security, vol. 10, no. 11, pp. 2396–2407, S. R. Arashloo, J. Kittler, and W. Christmas, "Face spoofing detection based on multiple descriptor fusion using multiscale dynamic binarized statistical image features."
[27] D. Menotti et al., “Deep representations for iris, face, and fingerprint spoofing detection,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 4, pp. 864–879, Apr. 2015.