Cleaner Adversarial CAPTCHAs: Intelligent Targets and Precise Noise for Usable Security

Meir Litman
Department of Industrial Engineering and Management
Ariel University
Ariel, Israel
meir.litman@msmail.ariel.ac.il

Chen Hajaj
Department of Industrial Engineering and Management
Ariel University
Ariel, Israel
chenha@ariel.ac.il

ABSTRACT

Traditional CAPTCHAs are increasingly vulnerable to deep learning-based solvers that decode text and images with high accuracy. In this work, we propose methods to strengthen adversarial CAPTCHAs without compromising human usability. First, we introduce a Precise Gradient Method (PGM) that preserves gradient magnitude (rather than discarding it via a sign operator), producing adversarial perturbations with significantly lower perceptual noise. Second, we develop intelligent target class selection, using either dataset-level confusion structure (Class Relations Network) or image-specific softmax probabilities (Distance-Based Target), to steer adversarial perturbations more efficiently. Across multiple modern architectures (MobileNets, EfficientNets, ResNet, and Vision Transformer), our framework achieves faster convergence (fewer iterations), reduced visual distortion, and notably greater robustness under iterative adversarial retraining. Experiments show that our methods consistently reduce iteration counts and perceptual distortion while significantly increasing the difficulty for automated attacks. Our results offer a practical, scalable path toward the next generation of CAPTCHA systems and contribute new insights to the adversarial machine learning landscape focused on security and usability.

KEYWORDS

Adversarial CAPTCHA, Usable Security, Intelligent Target Selection, Adversarial Robustness

ACM Reference Format:
Meir Litman and Chen Hajaj. 2026. Cleaner Adversarial CAPTCHAs: Intelligent Targets and Precise Noise for Usable Security. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/AFAW3962

1 INTRODUCTION

With the continuing expansion of online services, securing sensitive information is more critical than ever. Passwords remain the most widely used authentication mechanism, but they are increasingly vulnerable to brute-force attempts, phishing, and sophisticated automated attacks. To address these risks and differentiate between human users and malicious bots, websites commonly employ CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) [57].

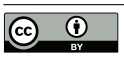
CAPTCHAs add an additional security layer by presenting challenges that are easily solvable for humans but highly resistant to automated algorithms. Over time, a wide range of CAPTCHA formats has been developed, including distorted text [57], perturbed images [11, 18, 21, 27], synthetically generated examples [2], and even audio tasks that require users to transcribe distorted speech [15]. Regardless of their modality, these mechanisms share a common design principle, producing samples that remain interpretable to human vision or hearing while posing substantial difficulty for automated systems. In adversarial ML terminology, these challenges can be viewed as adversarial examples intentionally crafted to exploit the asymmetry between human and machine perception.

However, designing effective CAPTCHA introduces a unique tension. In classical adversarial ML, adversarial examples are optimized solely to mislead machine learning (ML) models. CAPTCHA design, on the other hand, must satisfy an additional constraint, human readability. This tradeoff makes CAPTCHA generation a distinct and challenging problem in the adversarial ML landscape.

The scientific community has approached adversarial examples from two complementary perspectives. On one side, researchers attempt to defend ML networks by improving robustness [6, 48], enhancing their ability to correctly classify perturbed samples [35], or detecting adversarial inputs before misclassification occurs [37, 64]. On the other side, adversarial methods are leveraged to strengthen security mechanisms, such as CAPTCHA, by generating adversarial examples that humans can easily solve but ML algorithms cannot [7, 43].

Despite continuous innovations, the arms race between CAPTCHA designers and attackers has intensified. Well-known deep learning (DL) models [51] have become highly efficient image classifiers, enabling attackers to solve CAPTCHA at scale [4, 19, 55, 60]. Earlier adversarial CAPTCHA generation techniques each carried significant limitations: optimization-based approaches [54] were accurate but computationally expensive, while gradient-based methods such as the Fast Gradient Sign Method (FGSM) [20] were fast but fragile, often bypassable via simple preprocessing filters. DeepCAPTCHA [43] represented an important step forward by blending optimization strategies with efficient gradient methods, enabling real-time generation of adversarial CAPTCHAs that confused DL models while remaining legible to humans.

Yet challenges remain. As DL models become more robust year after year, generating effective adversarial CAPTCHA requires stronger perturbations. This not only increases execution time



Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/AFAW3962

but also risks producing images that are noisy or unpleasant from the human user's perspective. Addressing this trade-off—designing CAPTCHAs that remain both usable by humans and difficult for machines—is the problem that motivates this study.

2 OUR CONTRIBUTION

This paper makes the following key contributions to the design of secure and human-friendly adversarial CAPTCHAs:

(1) **Precise Gradient Method (PGM).** We introduce a novel adversarial perturbation technique that preserves gradient magnitude information rather than discarding it via the sign operator, as in FGSM. By distributing perturbations via a noise-spreading mechanism, PGM produces cleaner adversarial CAPTCHAs that significantly reduce perceptual distortion while maintaining human solvability.
(2) **Intelligent Target Class Selection.** To overcome the inefficiency of random target selection in prior work, we propose two complementary strategies:
   - **Class Relations Network (CRN):** A dataset-level structure that encodes frequent class confusions, enabling efficient adversarial generation by restricting deceiving classes to plausible alternatives.
   - **Distance-Based Target (DBT):** An image-specific approach that adaptively selects target classes from the classifier's probability distribution. By tuning the target distance, DBT steers perturbations toward classes that balance efficiency and imperceptibility.
(3) **Comprehensive Empirical Evaluation.** We conduct large-scale experiments on multiple modern architectures (MobileNetV2, MobileNetV3-Large, EfficientNet-B4, EfficientNet-B7, ResNet50, and ViT-B/16), covering both lightweight and heavyweight models. Our results show that PGM combined with CRN or DBT consistently reduces the number of required iterations and the magnitude of added noise compared to FGSM and random baselines.
(4) **Analysis of Robustness under Hardening.** We evaluate the resilience of the proposed strategies under adversarial retraining scenarios. The results demonstrate that intelligent target selection, particularly DBT, remains effective even as classifiers are hardened, underscoring the security relevance of the methods beyond naïve settings.

Both CRN and DBT minimize visible noise, thereby preserving human solvability while deceiving automated systems. We validate these methods on the large-scale ILSVRC dataset [49], which contains 1,000 object classes. Our findings confirm that the proposed framework enables rapid adversarial CAPTCHA generation suitable for modern, high-volume deployments, while maintaining strong robustness against automated attacks.

3 RELATED WORK

CAPTCHA [57] was introduced in 2003 as an automated test that humans can pass, but current computer programs can't. In the early stages, Text CAPTCHA was widely used, but according to more recent works [4, 19, 43], they are now almost useless or will become obsolete with advances in ML and especially DL. The breakthrough came in 2014 when Google's neural network outperformed humans at recognizing distorted text, originally designed as a purely human task. This led to a rapid transition from text-based to image-based CAPTCHA systems. For example, Atri et al. [3] achieved an average of 92% accuracy in detecting CAPTCHA text schemes.

Contemporary image CAPTCHAs, particularly reCAPTCHAv2, have also succumbed to advanced AI techniques. Studies from ETH Zurich show that AI using YOLO models can now solve 100% of reCAPTCHAv2 challenges, surpassing previous success rates of 68–71% [47]. This shift means AI bots now require roughly the same number of challenges as humans to pass CAPTCHA tests, undermining the premise of these security tests.

Later, audio and image CAPTCHAs were developed, where audio CAPTCHAs remain more challenging to solve than visual methods (texts, images) [5], and are typically reserved for visually impaired users [15, 17]. Image CAPTCHAs are often categorized as context-based and adversarial example-based. Context-based CAPTCHAs (such as CORTCHA [65] or [1]) present puzzles that require reasoning, whereas adversarial image CAPTCHAs rely on subtle perturbations unnoticeable to humans but that confuse ML algorithms.

Distinguishability measure [16] expresses how difficult it is for an ML model to classify an image. This concept explains common model failures with adversarial input [10]. Early classifiers (linear, quadratic) were limited in robustness; higher-capacity, non-linear models show more resilience but still face adversarial challenges.

Modern DL architectures learn highly non-linear functions but remain vulnerable. Adversarial examples usually transfer between models [44, 53], and transferability can be further enhanced [28, 34].

Several adversarial example creation techniques have emerged: Evolution Algorithms (EAs) [41] produce patterns that DNNs misclassify confidently; optimization techniques [20, 54] and localized distortion [44], or manipulation of internal representations [50].

Several noteworthy innovations concern universal adversarial perturbations (UAPs). DeepFool [38] introduced efficient computation of perturbations. Data-independent methods, such as those by [39], generate image-agnostic perturbations but tend to be more noticeable and less effective. Universal attacks on image spam [46], segmentation [25], and universal adversarial networks [24] demonstrate wide applicability [8, 40]. Recent advances include UAP generation via minimum-distance attacks on deep model decision boundaries [33], and Doubly-Universal Adversarial Perturbations (Doubly-UAP) for Vision-Language Models, which simultaneously target image and text inputs at attention layers [29]. Transferability remains central, with new frameworks operating data-agnostically and extensive comparative surveys [22].

Training networks on adversarial examples can improve generalization but remains insufficient for full robustness [16, 20]. Knowledge distillation [26], particularly defensive distillation [45], has shown some promise but is largely circumvented by modern attacks—defensive distillation offers limited protection and is vulnerable to parameter modifications [52]. Adversarial training is considered a leading defense [63]. Recent surveys categorize advancements in data augmentation, network design, and training configurations. Synthesized data [62] and sparse teacher methods [61] open new directions in balancing robustness and generalization. Autoencoder recovery [23] is also explored, but autoencoder+ML architectures may be vulnerable to bespoke attacks.

Work on adversarial CAPTCHAs continues to evolve. DeepCAPTCHA [43] aims to efficiently generate adversarial examples. New techniques seek to balance human/AI recognizability, creating either too-easy or too-difficult samples. Content Feature Alteration methods [58] aim to boost transferability of adversarial CAPTCHAs. The D-CAPTCHA system for deepfake defense employs adversarial training for increased robustness, reducing attack success rates [59]. Following this work, Nguyen-Le et al. [42] introduce a more robust version, D-CAPTCHA++, to defend against fake calls. Innovations include IllusionCAPTCHA [12], which exploits human perception of visual illusions unsolved by AI, and bi-phase adversarial frameworks [14], which blend multi-stage verification to maximize security.

4 ADVERSARIAL METHODS FOR CAPTCHA GENERATION

Fighting against DL algorithms that can reliably classify CAPTCHA images has become an active area of research in recent years [20, 54]. Image-based CAPTCHA generation typically relies on adversarial example generation techniques. Below, we review three representative approaches that form the foundation for modern adversarial CAPTCHA systems.

I. Optimization-Based Method [54]

This approach formulates adversarial example generation as a constrained optimization problem:

$$\arg\min_{\Delta_I} ||\Delta_I||^2 \quad \text{s.t. } Net(I + \Delta_I) = C_d, \tag{1}$$

where I is the original image, Δ_I is the perturbation, $Net(\cdot)$ is the DL classifier, and $C_d \neq C_i$ is the target deceiving class (different from the true class C_i).

The objective is to find the smallest perturbation Δ_I (in the ℓ_2 norm sense) that forces the classifier to mislabel the image. This method is highly effective and produces robust adversarial examples. However, since the optimization requires iterative search and the perturbations are not directionally guided, the process is computationally expensive and slow in practice.

II. Fast Gradient Sign Method (FGSM) [20]

The FGSM provides a more efficient alternative by generating perturbations in a single gradient step:

$$\Delta_I = \epsilon \cdot \text{sign}\big(\nabla_I J(W, I, C_i)\big), \tag{2}$$

where $J(W, I, C_i)$ is the loss function of the network parameterized by weights W, evaluated on input I with true class C_i. The perturbation magnitude is controlled by the hyperparameter ϵ.

Unlike the optimization method, FGSM does not require specifying a target class. Instead, the perturbed image is likely to be misclassified into some incorrect class. The method is computationally efficient and widely used for generating adversarial examples at scale. Nevertheless, it is less robust and less effective than the optimization-based approach.

An extension of FGSM is the Iterative FGSM (IFGSM) [30], in which Equation 2 is applied repeatedly in small steps. After each iteration, the adversarial example is re-evaluated by the classifier, and the process continues until misclassification occurs. IFGSM typically yields stronger adversarial examples at the cost of additional computation.

III. Immutable Adversarial Noise (IAN) [43]

IAN, introduced as part of the DeepCAPTCHA framework, integrates the principles of FGSM and IFGSM while explicitly targeting a deceiving class. The update rule is:

$$I^{t+1} = I^t - \epsilon \cdot \text{sign}\big(\nabla_I J(W, I^t, C_d)\big), \tag{3}$$

where I^t is the adversarial image at iteration t and C_d is the designated deceiving class. By using C_d instead of C_i, the method enforces targeted misclassification. The subtraction sign reflects the optimization direction toward aligning the input with the features of C_d.

To enhance robustness against filtering attacks, a median filter (typically 5×5) is applied to each generated adversarial example. If the filtered image is reclassified correctly as C_i, the perturbation magnitude ϵ is incremented ($\epsilon \leftarrow \epsilon +$ increment) and the process is repeated. Larger ϵ values correspond to stronger perturbations applied at each iteration.

While IAN improves resistance to filtering, its effectiveness depends on the DL model. For some modern architectures, adversarial examples targeting random deceiving classes may require many iterations before the classifier is successfully "convinced" to misclassify the input (see Figure 1). In our work, we adopt IAN as a baseline framework and propose updated core components to address these limitations.

IV. Fast Gradient Non-sign Method (FGNM) [9]

FGNM was proposed to mitigate the shortcomings of FGSM by addressing the directional bias introduced by the sign operator. The update rule is defined as:

$$I^{t+1} = I^t - \epsilon \cdot \frac{\nabla_I J(W, I^t, C_d)}{\|\nabla_I J(W, I^t, C_d)\|}, \tag{4}$$

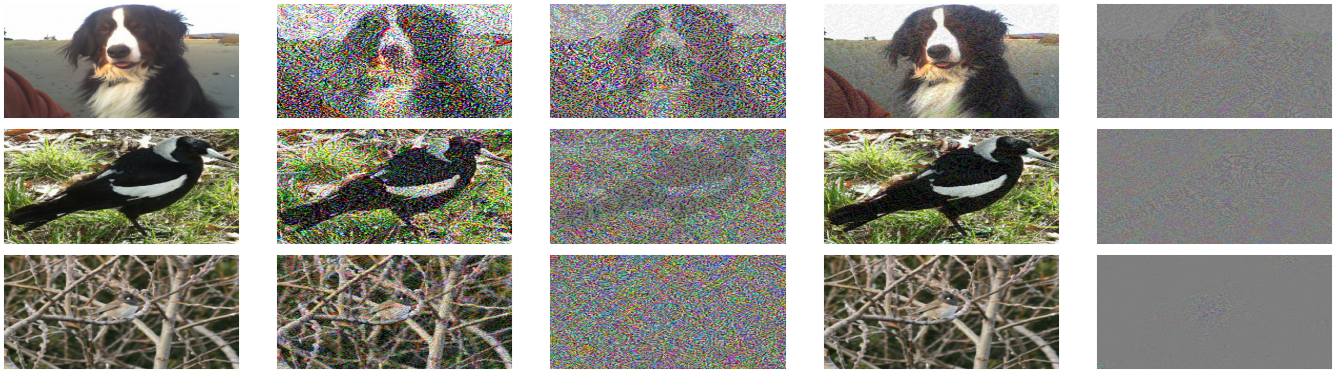
where the raw gradient is normalized by its norm before being applied. Unlike FGSM, which discards gradient magnitudes by taking only their signs, FGNM preserves this information, thereby achieving more precise perturbations.

Empirical evaluations [9] show that FGNM outperforms FGSM and several of its derivatives, such as DI-FGSM, MI-FGSM [13], and SI-FGSM [32]. While FGNM was designed to address the directional bias of FGSM by normalizing the raw gradients, our initial evaluations indicate that it performs worse than our proposed method when assessed under the L_∞ constraint. Therefore, we focus our analysis on FGSM and PGM, leaving FGNM for future research.

V. Precise Gradient Method (PGM)

FGSM generates noise that modifies nearly every pixel, since the sign operator does not distinguish between large and small gradient values. By contrast, optimization-based methods [54] tend to concentrate perturbations in salient regions of the image (see Figure 5 in [54]), but they are slow due to their iterative and often stochastic nature.

We propose the **Precise Gradient Method (PGM)**, combining FGSM's efficiency with optimization approaches' precision, while



(a) Original images (b) Perturbed images (FGSM) (c) FGSM perturbations (d) Perturbed images (PGM) (e) PGM perturbations

Figure 1: Visual comparison of FGSM and PGM on MobileNetV3. Column (a) shows originals. Columns (b) and (c) show FGSM perturbed images and their noise maps. Columns (d) and (e) show PGM perturbed images and their noise maps. PGM localizes noise to essential regions, producing cleaner, less perceptible adversarial samples.

preserving human readability. The key idea is to retain gradient magnitude information, which provides a finer control over the perturbation strength applied to each pixel. To prevent very small gradients from being neglected, we normalize them as follows:

$$NP = \frac{\nabla_I J(W, I, C_d)}{\max\left(|\nabla_I J(W, I, C_d)|\right)}, \tag{5}$$

where NP denotes the normalized noise pattern.

The adversarial examples generated using only NP are visually cleaner, but they exhibit susceptibility to median filtering (e.g., 5×5 filters). This vulnerability is reflected in a higher Filter Success Rate (FSR) of 14.07, compared to 3.97 for FGSM. To address this vulnerability, we introduce a Noise Spread (NS) component that redistributes perturbations more evenly over the image:

$$NS = \text{sign}(NP) - NP. \tag{6}$$

When combined with NP, the NS term significantly reduces the FSR, with average values dropping to 3.07.

The final adversarial perturbation is obtained as:

$$I_p = I - \left(\epsilon \cdot NP + \epsilon_s \cdot NS\right), \tag{7}$$

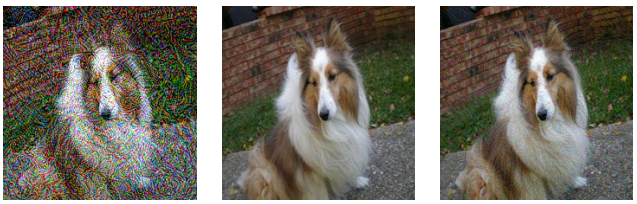
where I is the original image, I_p is the perturbed image, ϵ controls the magnitude of the normalized noise, and ϵ_s controls the contribution of the noise spread.

Unlike previous approaches that often overwhelm the image with excessive noise, PGM strategically introduces perturbations in specific regions while preserving the CAPTCHA's natural appearance. This balance between precision and efficiency makes PGM particularly suitable for practical adversarial CAPTCHA generation (see Figure 1 for a comparative visualization).

5 METHODS

Advances in image classification models pose a serious challenge to the effectiveness of traditional CAPTCHAs. Recent work, such as the Image Adversarial Network (IAN) [43], attempts to generate adversarial CAPTCHAs that deceive state-of-the-art (SOTA) classifiers. However, IAN-generated CAPTCHAs frequently suffer from a critical drawback: the perturbations required to achieve misclassification are often visually noticeable to human users. This compromises the practical usability of the CAPTCHA, since its purpose is to remain solvable for humans while resisting automated attacks.

Figure 2 illustrates this limitation. When perturbing an image of a Shetland Shepherd with the goal of having it misclassified as a *sea lion*, a substantial amount of noise must be introduced ($\epsilon = 0.145$). The resulting adversarial CAPTCHA is strongly manipulated, with distortions that are easily perceptible to human observers.



(a) Perturbed (sea lion). $\epsilon = 0.145$, **175 iterations** (b) Original image. $\epsilon = 0$ (c) Perturbed (hognose). $\epsilon = 0.025$, **76 iterations**

Figure 2: Effect of target class selection on noise levels. Large visible noise when targeting "sea lion" ($\epsilon = 0.145$). The original, unperturbed image. A cleaner perturbation when targeting "hognose snake", requiring fewer iterations (76) and introducing less noise ($\epsilon = 0.025$).

By contrast, selecting a more compatible deceiving class reduces the required perturbation. For instance, targeting the class *hognose snake* produces a successful misclassification after 76 iterations with significantly less noise ($\epsilon = 0.025$). This example highlights the strong dependence of adversarial CAPTCHA generation on the choice of target class: inappropriate targets lead to excessive noise, while more suitable targets yield cleaner adversarial examples.

The presence of visible noise in IAN-generated CAPTCHAs not only hampers their effectiveness by making image classification

more challenging for humans, but also risks producing false positives where legitimate users are misclassified as automated bots. Furthermore, adversaries can exploit these noise patterns by applying simple filtering techniques, such as median filters, to suppress the perturbations and bypass the CAPTCHA altogether. These limitations underscore the need for adversarial CAPTCHA techniques that minimize perceptible noise, ensuring that generated CAPTCHAs remain indistinguishable from the original images to human observers.

In this paper, we propose two key improvements to the IAN methodology. First, we introduce the **Precise Gradient Method (PGM)** as a replacement for the Fast Gradient Sign Method (FGSM). By leveraging full gradient information rather than only their signs, PGM produces cleaner and more effective adversarial examples, significantly reducing visible perturbations. Second, we propose an **intelligent target class selection strategy** that replaces random target assignment with a principled, classifier-informed approach. This strategy improves both the speed of adversarial example generation and the perceptual quality of the resulting CAPTCHAs. By explicitly incorporating the perceptibility factor into the generation process, our approach strikes a balance between deceiving the classifier and preserving visual integrity for human users.

A central element of our approach is the development of an intelligent strategy for selecting target classes. Instead of choosing the deceiving class C_d randomly, we exploit the probability vector produced by multiple pre-trained classifiers. Without loss of generality, in our work, each classifier is trained on the ILSVRC2012 validation dataset. When aiming to choose the deciving calls, it takes an input image and produces a probability distribution over the 1,000 classes introduced in this dataset. This distribution can be represented as:

$$C \in \mathbb{R}^{1000}, \quad C_i = P(\text{class } i \mid I),$$

where C_i denotes the predicted probability that the image I belongs to class i. Typically, the correct class C_{true} will have the highest predicted probability.

To formalize target selection, we sort the probability vector in ascending order:

$$C^s = \text{sort}(C),$$

where C^s_0 corresponds to the least likely class and C^s_{999} to the most probable class (usually the correct label). For example, if C^s_{999} has the highest probability, it almost certainly corresponds to the ground-truth class. By contrast, C^s_{998} and C^s_{997} represent classes with progressively lower probabilities, indicating that they are less likely to be the correct class. Finally, C^s_0 is the least likely class according to the classifier.

This probability-based ranking enables us to design selection strategies that avoid random guessing. For instance, by choosing deceiving classes from among low- to mid-probability candidates (rather than the extremes), we can identify targets that simultaneously require less noise to achieve misclassification and maintain higher visual fidelity of the CAPTCHA.

5.1 Class Relations Network (CRN)

To further accelerate and improve the adversarial CAPTCHA generation process, we introduce the Class Relations Network (CRN). The CRN captures structured co-occurrence patterns between classes, derived from the predictions of a large-scale image classifier.

Using ILSVRC2012 validation set [49], which contains 50 images for each of 1,000 classes, we constructed a 1000×1000 co-occurrence matrix by recording the top-5 predicted classes for each image and then extracting, for every class, the 20 most frequently co-occurring alternatives. This yields a compact structure in which each class is associated with its most likely "confusion partners", reflecting relationships in the learned feature space (see Table 1).

Table 1: Class relations Network for the first ten classes. The rows represent the classes (0-999), while the columns represent each prediction's closeness to that class's images.

Class	Prediction									
	1	2	3	4	5	6	7	8	9	10
0	0	389	391	955	758	1	395	394	997	29
1	1	393	29	0	27	392	108	115	397	973
2	2	3	4	148	394	147	389	149	5	6
3	3	2	4	149	394	6	5	983	148	150
4	4	3	2	394	6	983	33	150	148	5
5	5	6	397	390	33	4	109	110	115	329
...										
994	994	947	997	992	995	993	991	63	114	124
995	995	994	997	991	947	988	993	996	992	990
996	996	991	947	993	994	995	997	953	825	961
997	997	947	992	995	994	996	993	990	988	666
998	998	987	954	398	955	122	599	939	943	953
999	999	700	434	876	896	281	435	591	606	617

For example, for images belonging to class 0, the class itself naturally appeared most frequently among the top-5 predictions. In addition, class 391 was predicted more often than class 955, yet less frequently than class 389. This ranking reflects how closely related these classes appear in the feature space learned by the classifier.

Rather than assigning deceiving classes randomly, CRN restricts the target class to one of these plausible alternatives. This design ensures that perturbations push the classifier toward a class that is already semantically or visually related to the true label, which reduces the number of required iterations and the magnitude of noise. At the same time, if the chosen class is too close to the original label (e.g., among the top-5 confusions), the perturbations may become trivial or uninformative, while overly distant classes reintroduce high distortion.

To balance efficiency and perceptual quality, we therefore evaluate CRN using target classes ranked at the 10th most frequent co-occurrences. These mid-ranked classes are sufficiently distinct from the original label to produce meaningful adversarial shifts, while still close enough to allow efficient generation with minimal visible distortion. In Section VI, we empirically confirm this trade-off by comparing results at the 10th and 20th ranks.

5.2 Distance-based Target (DBT)

The Distance-Based Target (DBT) method selects a deceiving class directly from the probability distribution produced by the classifier for each input image. Instead of randomly choosing a target class, DBT leverages the classifier's uncertainty: it sorts the output probabilities and then selects one of the alternative classes as

the perturbation target. This adaptive choice makes DBT image-specific, ensuring that the perturbation exploits local ambiguities in the classifier's decision space.

An important design decision is which rank in the sorted probability vector to use. Selecting a class that is too close to the true label (e.g., the second most probable class, "distance 1") often results in trivial perturbations. These small changes can successfully fool the model but are visually confusing or unstable, making them less suitable for CAPTCHAs. Conversely, choosing a class that is too far from the true label (e.g., a distance of 100 or more) requires large perturbations, which produce noticeable visual distortion and degrade human usability.

To strike a balance, we evaluate DBT across multiple distances, ranging from 1 to 100. Distances in the near-to-mid range (e.g., 5–20) are expected to provide the best trade-off: sufficiently distinct from the true label to generate meaningful adversarial shifts, yet not so distant that perturbations become excessive. In the following section, we empirically test this design choice by comparing iteration counts and perturbation magnitudes across distances 1, 5, 10, 20, 50, and 100.

Table 2: Comparison of average iterations required for successful adversarial example generation using Class Relations Network (CRN) versus random target selection across several architectures. Values show mean and standard deviation over all test samples.

Model (Attack)	Random	CRN (10th)
MobileNetV2 (FGSM)	3.18 (0.61)	1.79 (0.29)
MobileNetV2 (PGM)	2.64 (0.25)	1.55 (0.21)
MobileNetV3-L (FGSM)	4.02 (0.55)	2.41 (0.37)
MobileNetV3-L (PGM)	3.33 (0.41)	2.09 (0.33)
ResNet50 (FGSM)	3.61 (0.72)	2.26 (0.42)
ResNet50 (PGM)	2.95 (0.51)	1.84 (0.33)
EfficientNet-B4 (FGSM)	6.61 (0.88)	4.12 (0.66)
EfficientNet-B4 (PGM)	5.38 (0.72)	3.27 (0.53)
EfficientNet-B7 (FGSM)	4.81 (0.67)	2.85 (0.45)
EfficientNet-B7 (PGM)	3.92 (0.56)	2.29 (0.39)
ViT-B/16 (FGSM)	4.59 (2.42)	2.50 (0.97)
ViT-B/16 (PGM)	4.50 (1.36)	2.47 (0.99)

6 EXPERIMENTAL RESULTS

We evaluate six architectures (i.e., MobileNetV2, MobileNetV3-Large, ResNet50, EfficientNet-B4, EfficientNet-B7, and ViT-B/16) on the ImageNet validation set, using FGSM and PGM attacks under a shared perturbation schedule ($\epsilon = 0.005$, incremented by 0.005 per iteration). Metrics include the mean number of iterations to success and L2 distortion. The experiments were conducted on a workstation equipped with an AMD Ryzen Threadripper PRO 5955WX processor. The system was configured with 64 GB of DDR4 RAM and an NVIDIA RTX A5000 GPU with 4 GB of GDDR6 video memory, providing sufficient computational resources for training and evaluating deep learning models in our study.

CRN effectiveness. Our first experiment evaluates the effectiveness of CRN by comparing it with random selection as a baseline. Table 2 reports the iteration counts across multiple architectures under both FGSM and PGM attacks. The results consistently favor CRN across all tested models, as CRN converges faster than random selection. For example, EfficientNet-B7 with FGSM requires on average 4.81 iterations under random selection, but only 2.85 with CRN, corresponding to a $\approx 41\%$ reduction. Similar improvements are observed across lightweight models such as MobileNetV2 and deeper architectures such as EfficientNet-B4, confirming the robustness of this effect. These findings highlight CRN's role as an effective dataset-level heuristic that accelerates adversarial generation in a structured and reliable manner.

DBT distance trade-offs. We next examined the effect of varying DBT distance ranks on iteration counts for FGSM across distances from 1 to 100. For convolutional models, all results follow a consistent monotonic pattern: the smallest distances (ranks 1–5) yield the fastest adversarial convergence, mid-range distances (10–20) balance efficiency with perceptual plausibility, and very large distances (50–100) result in slower convergence, often approaching random baselines. Interestingly, the Vision Transformer (ViT) diverges from this trend, showing less sensitivity to distance rank and weaker monotonicity. This suggests that ViTs may distribute decision boundaries differently from convolutional architectures, making the DBT parameter less effective as a fine-grained control knob. These findings highlight that while DBT distance is a powerful and tunable parameter for CNN-based models, its utility may be reduced when targeting transformer-based architectures.

PGM versus FGSM. To comprehensively evaluate the relative efficiency and subtlety of FGSM and PGM attack methods, Table 3 reports their iteration counts under different selector strategies. Across all architectures, PGM converges faster than FGSM, with the gap widening when more informed selectors, such as CRN-10 and DBT-10, are used. For example, on MobileNetV2 with DBT-10, PGM reduces the mean number of iterations from 2.11 with FGSM to 1.64. This systematic improvement highlights PGM's superior convergence efficiency, confirming that gradient normalization combined with spread adjustment enables quicker and more reliable adversarial generation.

Supporting these findings, Table 4 shows that PGM consistently generates adversarial perturbations with lower L2 distortion, often reducing noise by 20–40% across models and selectors. For example, under random selection on EfficientNet-B4, PGM achieves an average L2 norm of 47.17 compared to 67.70 for FGSM. Similar reductions are observed on MobileNet, ResNet, and ViT, confirming that PGM achieves a better trade-off between convergence speed and imperceptibility across architectures.

Although absolute iteration counts vary by architecture, with EfficientNets typically requiring more iterations than MobileNets, the relative ordering of methods is stable. Across our evaluations, both CRN and DBT consistently outperform random baselines, and PPGM reliably outperforms FGSM in convergence speed and perceptual noise reduction. We note an important caveat, ViT-B/16 exhibits different sensitivity to DBT distance ranks than convolutional models, indicating that target selection strategies may need to be adapted for transformer architectures.

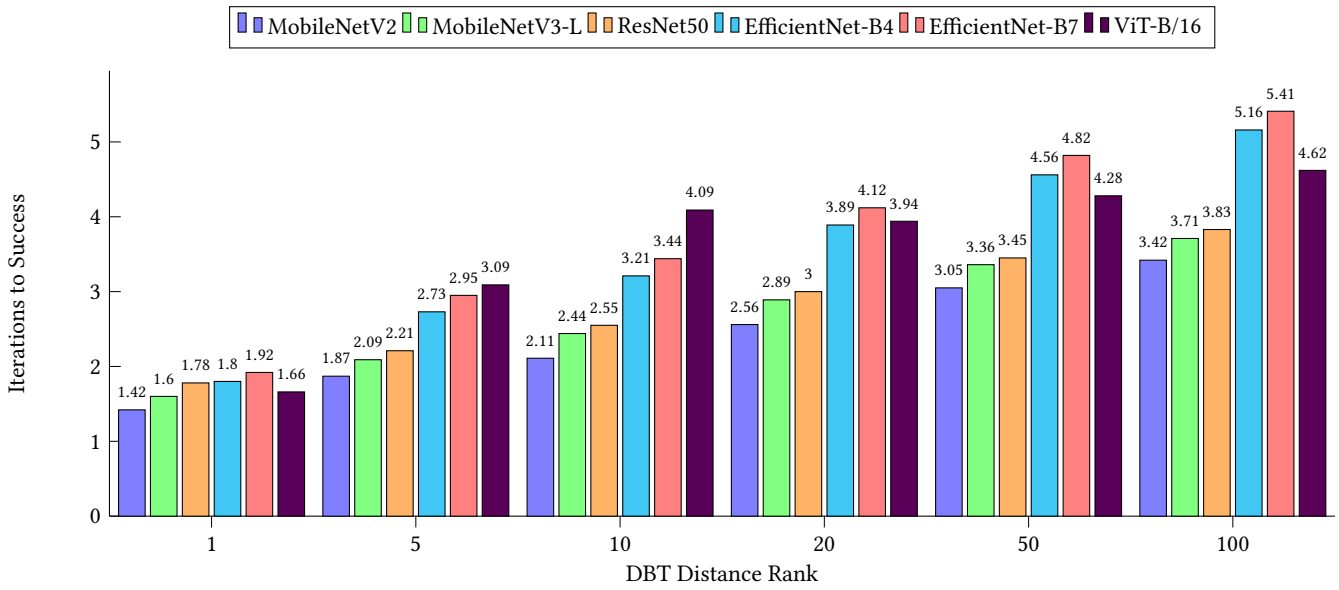


Figure 3: Effect of varying Distance-Based Target (DBT) rank distance on iterations required to generate successful adversarial examples using FGSM. Bars represent mean iterations for each convolutional or transformer architecture evaluated across distance ranks 1, 5, 10, 20, 50, and 100.

Table 3: Comparison of average iterations to success between PGM and FGSM across different target selection strategies (Random, CRN-10th, DBT-10). Results are shown for multiple architectures, with values as mean (std.).

Model	Selector	FGSM	PGM
MobileNetV2	Random	3.18 (0.61)	2.64 (0.25)
	CRN-10th	1.79 (0.29)	1.55 (0.21)
	DBT-10	2.11 (0.26)	1.64 (0.22)
MobileNetV3-L	Random	4.02 (0.55)	3.33 (0.41)
	CRN-10th	2.41 (0.37)	2.09 (0.33)
	DBT-10	2.44 (0.32)	1.98 (0.28)
ResNet50	Random	3.61 (0.72)	2.95 (0.51)
	CRN-10th	2.26 (0.42)	1.84 (0.33)
	DBT-10	2.55 (0.36)	2.04 (0.31)
EfficientNet-B4	Random	6.61 (0.88)	5.38 (0.72)
	CRN-10th	4.12 (0.66)	3.27 (0.53)
	DBT-10	3.21 (0.44)	2.48 (0.38)
EfficientNet-B7	Random	4.81 (0.67)	3.92 (0.56)
	CRN-10th	2.85 (0.45)	2.29 (0.39)
	DBT-10	3.44 (0.47)	2.71 (0.41)
ViT-B/16	Random	5.10 (0.72)	4.08 (0.61)
	CRN-10th	3.27 (0.48)	2.65 (0.42)
	DBT-10	3.62 (0.53)	2.89 (0.47)

Table 4: Mean L2 distortion norms (with standard deviation) of adversarial perturbations generated by FGSM and PGM under various target selection methods across multiple architectures.

Model	Selector	FGSM	PGM
MobileNetV2	Random	24.05 (3.91)	16.96 (2.17)
	CRN-10th	18.32 (3.44)	12.44 (1.98)
	DBT-10	20.45 (3.67)	14.07 (2.06)
MobileNetV3-L	Random	35.47 (4.21)	23.73 (2.85)
	CRN-10th	26.18 (3.89)	17.93 (2.44)
	DBT-10	28.09 (4.02)	19.54 (2.56)
ResNet50	Random	32.88 (4.08)	21.63 (2.71)
	CRN-10th	26.32 (3.77)	17.08 (2.39)
	DBT-10	28.74 (3.95)	18.92 (2.51)
EfficientNet-B4	Random	67.70 (5.31)	47.17 (4.94)
	CRN-10th	54.62 (5.02)	35.44 (4.38)
	DBT-10	58.21 (5.13)	39.02 (4.65)
EfficientNet-B7	Random	48.35 (4.89)	36.24 (4.12)
	CRN-10th	36.12 (4.42)	26.98 (3.84)
	DBT-10	39.81 (4.56)	29.62 (3.97)
ViT-B/16	Random	42.71 (4.66)	31.54 (3.87)
	CRN-10th	33.28 (4.12)	24.79 (3.51)
	DBT-10	36.02 (4.28)	26.91 (3.65)

7 EVALUATING ROBUSTNESS AGAINST ADVERSARIAL RETRAINING

Adversarial retraining has emerged as one of the most widely studied defenses, where the classifier is incrementally exposed to adversarial examples in order to improve robustness [31, 36, 56]. We simulated five phases of such retraining on a MobileNetV2 classifier by adding 10,000 newly perturbed images per phase. This process broadens the training distribution to include adversarial samples, but also risks biasing the classifier toward perturbed inputs at the expense of clean-image performance. Figure 4 shows the average number of iterations required to generate successful adversarial examples across retraining phases. Three clear patterns emerge:

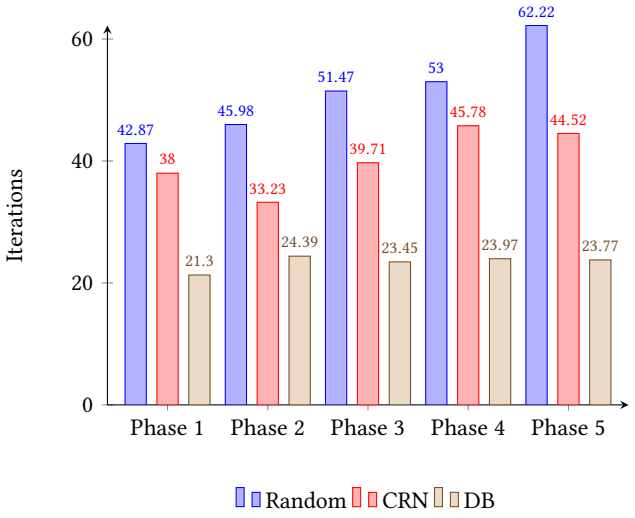


Figure 4: Average iterations to generate successful adversarial examples on MobileNetV2 during five phases of adversarial retraining.

(1) The **random baseline**, using IAN-style random target selection, becomes significantly harder to fool as iteration counts steadily increase across phases.
(2) The **Class Relations Network** exhibits much greater resilience: while its iteration counts fluctuate, the increases are marginal compared to the random baseline.
(3) The **Distance-Based Target** remains remarkably stable, with iteration requirements essentially flat across all five phases. This highlights DBT's capacity to preserve efficiency even against hardened classifiers.

To quantify the trade-off with clean inputs, Table 5 reports AUC values on unperturbed ILSVRC2012 validation images. Across all strategies, retraining introduces a steady decline in clean-image AUC, confirming that greater robustness against adversarial perturbations comes at the cost of reduced natural accuracy. This effect is especially visible for random selection, which suffers the sharpest drop, whereas CRN and DBT experience smaller degradations.

Table 5: Clean-image classification AUC of MobileNetV2 across adversarial retraining phases.

Method	0 (initial)	1	2	3	4	5
Random	0.971	0.961	0.956	0.953	0.950	0.945
CRN	0.971	0.965	0.961	0.955	0.951	0.947
DB	0.971	0.962	0.957	0.955	0.951	0.950

These findings emphasize two key insights. From the attacker's perspective, informed target selection strategies such as CRN and DBT substantially mitigate the impact of adversarial retraining, ensuring stable attack efficiency even as defenses evolve. From the defender's perspective, adversarial retraining cannot be applied indiscriminately, as its benefits against perturbed inputs are offset by weakened performance on clean data. This trade-off underscores the adversarial "arms race" dynamic: while retraining raises the computational bar for naïve attacks, structured strategies like DBT retain their effectiveness, highlighting the need for defenders to combine retraining with complementary defenses such as clean-sample reinforcement or hybrid detection systems.

8 DISCUSSION AND CONCLUSIONS

This research set out to address a fundamental question: how can adversarial CAPTCHA generation balance security, usability, and scalability in an era where machine learning–based solvers are increasingly capable? Traditional CAPTCHAs either impose excessive cognitive burden on users or fail to withstand automated attacks. We hypothesized that by combining gradient-aware perturbation methods with principled target class selection, it is possible to generate CAPTCHAs that remain solvable for humans while resisting automated systems, even at web-scale deployment.

Our findings confirm this intuition. The proposed Precise Gradient Method (PGM), together with Class Relations Network (CRN) and Distance-Based Target (DBT) target selectors, consistently reduces both iteration counts and perceptual distortion compared to baselines such as FGSM. CRN leverages dataset-level confusions to generate faster, cleaner adversarial examples, while DBT adaptively exploits classifier uncertainty to minimize distortion. These benefits translate into real-world potential: lower latency, reduced visual burden on users, and the ability to generate adversarial CAPTCHAs in real time for millions of web transactions. A specific finding stands out. CRN-based target selection exploits dataset-level confusions to achieve faster convergence with lower noise, while DBT leverages classifier-specific uncertainty to adaptively steer perturbations. In CNN-based architectures, DBT shows a clear monotonic relationship between class rank distance and adversarial difficulty, with near-targets requiring fewer iterations. Interestingly, this pattern breaks in transformer-based models such as ViT-B/16, where global attention and flatter probability distributions weaken the link between rank distance and semantic similarity. This suggests that target selection strategies are architecture-dependent, a key consideration for future CAPTCHA design.

Despite these advances, limitations remain. Adaptive adversaries may counteract perturbation-based CAPTCHAs through defenses such as denoising, compression, or adversarial retraining. Although our experiments show that retraining can degrade clean accuracy while only partially mitigating targeted attacks, the broader arms race between CAPTCHA designers and solvers persists. Moreover, the non-monotonic behavior of DBT in ViTs highlights the challenge of ensuring predictable performance across diverse architectures. Looking forward, future work should consider hybrid approaches that integrate perturbation-based CAPTCHA systems with behavioral or contextual signals, as seen in modern reCAPTCHA systems. Expanding to multimodal challenges (e.g., incorporating audio and text) offers another path to enhance robustness while preserving human accessibility. Finally, adaptive schemes that evolve target selection in response to solver feedback may help maintain long-term resilience.

REFERENCES

[1] Firkhan Ali Bin Hamid Ali and Farhana Bt Karim. 2014. Development of CAPTCHA system based on puzzle. In *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*. IEEE, 426–428.

[2] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Synthesizing robust adversarial examples. In *International conference on machine learning*. PMLR, 284–293.

[3] Aditya Atri, Ankita Bansal, Manju Khari, and S Vimal. 2022. De-CAPTCHA: A novel DFS based approach to solve CAPTCHA schemes. *Computers & Electrical Engineering* 97 (2022), 107593.

[4] Elie Bursztein, Jonathan Aigrain, Angelika Moscicki, and John C Mitchell. 2014. The end is nigh: Generic solving of text-based CAPTCHAs. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*.

[5] Elie Bursztein, Steven Bethard, Celine Fabry, John C Mitchell, and Dan Jurafsky. 2010. How good are humans at solving CAPTCHAs? A large scale evaluation. In *2010 IEEE symposium on security and privacy*. IEEE, 399–413.

[6] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705* (2019).

[7] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. Ieee, 39–57.

[8] Ashutosh Chaubey, Nikhil Agrawal, Kavya Barnwal, Keerat K Guliani, and Pramod Mehta. 2020. Universal adversarial perturbations: A survey. *arXiv preprint arXiv:2005.08087* (2020).

[9] Yaya Cheng, Xiaosu Zhu, Qilong Zhang, Lianli Gao, and Jingkuan Song. 2021. Fast Gradient Non-sign Methods. *arXiv preprint arXiv:2110.12734* (2021).

[10] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. 2004. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 99–108.

[11] Ritendra Datta, Jia Li, and James Z Wang. 2005. Imagination: a robust image-based captcha generation system. In *Proceedings of the 13th annual ACM international conference on Multimedia*. 331–334.

[12] Ziqi Ding, Gelei Deng, Yi Liu, Junchen Ding, Jieshan Chen, Yulei Sui, and Yuekang Li. 2025. IllusionCAPTCHA: A CAPTCHA based on visual illusion. (2025), 3683–3691.

[13] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Xiaolin Hu, and Jun Zhu. 2017. Discovering adversarial examples with momentum. *arXiv preprint arXiv:1710.06081* (2017).

[14] Xia Du, Xiaoyuan Liu, Jizhe Zhou, Zheng Lin, Chi-man Pun, Zhe Chen, Wei Ni, and Jun Luo. 2025. Unsourced Adversarial CAPTCHA: A Bi-Phase Adversarial CAPTCHA Framework. *arXiv preprint arXiv:2506.10685* (2025).

[15] Valerie Fanelle, Sepideh Karimi, Aditi Shah, Bharath Subramanian, and Sauvik Das. 2020. Blind and Human: Exploring More Usable Audio {CAPTCHA} Designs. In *Sixteenth Symposium on Usable Privacy and Security (SOUPS 2020)*. 111–125.

[16] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2018. Analysis of classifiers' robustness to adversarial perturbations. *Machine learning* 107, 3 (2018), 481–508.

[17] Haichang Gao, Honggang Liu, Dan Yao, Xiyang Liu, and Uwe Aickelin. 2010. An audio CAPTCHA to distinguish humans from computers. In *2010 Third International Symposium on Electronic Commerce and Security*. IEEE, 265–269.

[18] Haichang Gao, Dan Yao, Honggang Liu, Xiyang Liu, and Liming Wang. 2010. A novel image based CAPTCHA using jigsaw puzzle. In *2010 13th IEEE International Conference on Computational Science and Engineering*. IEEE, 351–356.

[19] Ian Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. 2014. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. In *ICLR2014*.

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. CoRR abs/1412.6572 (2014). *arXiv preprint arXiv:1412.6572* (2014).

[21] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. 2009. What's up CAPTCHA? A CAPTCHA based on image orientation. In *Proceedings of the 18th international conference on World wide web*. 841–850.

[22] Jindong Gu, Xiaojun Jia, Pau de Jorge, Wenqain Yu, Xinwei Liu, Avery Ma, Yuan Xun, Anjun Hu, Ashkan Khakzar, Zhijiang Li, et al. 2023. A survey on transferability of adversarial examples across deep neural networks. *arXiv preprint arXiv:2310.17626* (2023).

[23] Shixiang Gu and Luca Rigazio. 2014. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068* (2014).

[24] Jamie Hayes and George Danezis. 2018. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 43–49.

[25] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. 2017. Universal adversarial perturbations against semantic image segmentation. In *Proceedings of the IEEE international conference on computer vision*. 2755–2764.

[26] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).

[27] Dorjan Hitaj, Briland Hitaj, Sushil Jajodia, and Luigi Vincenzo Mancini. 2020. Capture the Bot: Using Adversarial Examples to Improve CAPTCHA Robustness to Bot Attacks. *IEEE Intelligent Systems* (2020).

[28] Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. 2019. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4733–4742.

[29] Hee-Seon Kim, Minbeom Kim, and Changick Kim. 2024. Doubly-universal adversarial perturbations: Deceiving vision-language models across both images and text with a single perturbation. *arXiv preprint arXiv:2412.08108* (2024).

[30] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[31] Bo Li, Yevgeniy Vorobeychik, and Xinyun Chen. 2016. A general retraining framework for scalable adversarial classification. *arXiv preprint arXiv:1604.02606* (2016).

[32] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. 2019. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv preprint arXiv:1908.06281* (2019).

[33] Dengbo Liu, Zhi Li, and Daoyun Xu. 2025. Generate universal adversarial perturbations by shortest-distance soft maximum direction attack. *Computers & Security* 150 (2025), 104168.

[34] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770* (2016).

[35] Jiajun Lu, Theerasit Issaranon, and David Forsyth. 2017. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE international conference on computer vision*. 446–454.

[36] Nag Mani, Melody Moh, and Teng-Sheng Moh. 2021. Defending deep learning models against adversarial attacks. *International Journal of Software Science and Computational Intelligence (IJSSCI)* 13, 1 (2021), 1–18.

[37] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. 2017. On Detecting Adversarial Perturbations. In *International Conference on Learning Representations*.

[38] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.

[39] KR Mopuri, U Garg, and R Venkatesh Babu. 2017. Fast feature fool: A data independent approach to universal adversarial perturbations. In *British Machine Vision Conference 2017, BMVC 2017*. BMVA Press.

[40] Chaithanya Kumar Mummadi, Thomas Brox, and Jan Hendrik Metzen. 2019. Defending against universal perturbations with shared adversarial training. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4928–4937.

[41] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 427–436.

[42] Hong-Hanh Nguyen-Le, Van-Tuan Tran, Dinh-Thuc Nguyen, and Nhien-An Le-Khac. 2024. D-captcha++: A study of resilience of deepfake captcha under transferable imperceptible adversarial attack. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[43] Margarita Osadchy, Julio Hernandez-Castro, Stuart Gibson, Orr Dunkelman, and Daniel Pérez-Cabo. 2017. No bot expects the DeepCAPTCHA! Introducing immutable adversarial examples, with applications to CAPTCHA generation. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2640–2653.

[44] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.

[45] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 582–597.

[46] Andy Phung and Mark Stamp. 2021. Universal adversarial perturbations and image spam classifiers. *Malware Analysis Using Artificial Intelligence and Deep Learning* (2021), 633–651.

[47] Andreas Plesner, Tobias Vontobel, and Roger Wattenhofer. 2024. Breaking recaptchav2. (2024), 1047–1056.

[48] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 2018. Certified Defenses against Adversarial Examples. In *International Conference on Learning Representations*.

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[50] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. 2015. Adversarial manipulation of deep representations. *arXiv preprint arXiv:1511.05122* (2015).

[51] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[52] Marcus Soll, Tobias Hinz, Sven Magg, and Stefan Wermter. 2019. Evaluating Defensive Distillation for Defending Text Processing Neural Networks against Adversarial Examples. In *Proceedings of the 15th International Conference on Security and Cryptography*. 685–692.

[53] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[54] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*.

[55] Sheng Tian and Tao Xiong. 2020. A generic solver combining unsupervised learning and representation learning for breaking text-based captchas. In *Proceedings of The Web Conference 2020*. 860–871.

[56] Liang Tong, Bo Li, Chen Hajaj, Chaowei Xiao, Ning Zhang, and Yevgeniy Vorobeychik. 2019. Improving robustness of ML classifiers against realizable evasion attacks using conserved features. In *28th USENIX Security Symposium (USENIX Security 19)*. 285–302.

[57] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. 2003. CAPTCHA: Using hard AI problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 294–311.

[58] Zisheng Xu and Qiao Yan. 2024. Boosting the transferability of adversarial CAPTCHAs. *Computers & Security* 145 (2024), 104000.

[59] Lior Yasur, Guy Frankovits, Fred M Grabovski, and Yisroel Mirsky. 2023. Deepfake captcha: A method for preventing fake calls. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 608–622.

[60] Guixin Ye, Zhanyong Tang, Dingyi Fang, Zhanxing Zhu, Yansong Feng, Pengfei Xu, Xiaojiang Chen, and Zheng Wang. 2018. Yet another text captcha solver: A generative adversarial network based approach. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 332–348.

[61] Eda Yilmaz and Hacer Yalim Keles. 2025. Adversarial sparse teacher: Defense against distillation-based model stealing attacks using adversarial examples. *IEEE Access* (2025).

[62] Li Zhang, Jun Wang, and Ming Chen. 2024. Adversarial Training with Synthesized Data: A Path to Robust Learning. *OpenReview* (2024). Available online at https://openreview.net/forum?id=H6V1NW7bGS.

[63] Mengnan Zhao, Lihe Zhang, Jingwen Ye, Huchuan Lu, Baocai Yin, and Xinchao Wang. 2024. Adversarial training: A survey. *arXiv preprint arXiv:2410.15042* (2024).

[64] Pinlong Zhao, Zhouyu Fu, Qinghua Hu, Jun Wang, et al. 2018. Detecting adversarial examples via key-based network. *arXiv preprint arXiv:1806.00580* (2018).

[65] Bin B Zhu, Jeff Yan, Qiujie Li, Chao Yang, Jia Liu, Ning Xu, Meng Yi, and Kaiwei Cai. 2010. Attacks and design of image recognition CAPTCHAs. In *Proceedings of the 17th ACM conference on Computer and communications security*. 187–200.