PROACTIVE ADVERSARIAL DEFENSE: HARNESSING PROMPT TUNING IN VISION-LANGUAGE MODELS TO DETECT UNSEEN BACKDOORED IMAGES

Kyle Stein[1], *Andrew A. Mahyari*[1,2], *Guillermo Francia, III*[3], *Eman El-Sheikh*[3]

[1] Department of Intelligent Systems and Robotics, University of West Florida, Pensacola, FL, USA
[2] Florida Institute For Human and Machine Cognition (IHMC), Pensacola, FL, USA
[3] Center for Cybersecurity, University of West Florida, Pensacola, FL, USA

ABSTRACT

Backdoor attacks pose a critical threat by embedding hidden triggers into inputs, causing models to misclassify them into target labels. While extensive research has focused on mitigating these attacks in object recognition models through weight fine-tuning, much less attention has been given to detecting backdoored samples directly. Given the vast datasets used in training, manual inspection for backdoor triggers is impractical, and even state-of-the-art defense mechanisms fail to fully neutralize their impact. To address this gap, we introduce a groundbreaking method to detect unseen backdoored images during both training and inference. Leveraging the transformative success of prompt tuning in Vision Language Models (VLMs), our approach trains learnable text prompts to differentiate clean images from those with hidden backdoor triggers. Experiments demonstrate the exceptional efficacy of this method, achieving an impressive average accuracy of 86% across two renowned datasets for detecting unseen backdoor triggers, establishing a new standard in backdoor defense.

***Index Terms*—Adversarial attacks, Backdoor, Vision-Language Model, Prompt tuning**

I. INTRODUCTION

Deep neural networks (DNNs) have revolutionized fields ranging from object classification [1] and face recognition [2] to reinforcement learning [3] and natural language processing [4], setting new benchmarks in performance and innovation. However, this remarkable success has made them prime targets for sophisticated adversarial manipulations. Among the most insidious threats are backdoor attacks, which stealthily embed hidden patterns—known as triggers—into models, causing them to misclassify inputs into an adversary's chosen target label. These backdoors can be implanted through malicious techniques like data poisoning [5] or neuron hijacking [6], posing an immediate and formidable challenge. In response, the research community has developed numerous defense and detection strategies [7], [8], [9], [10], [11]. Early approaches focused on purifying compromised models using methods such as fine-tuning [12], [13] or distillation [14]. More recently, cutting-edge techniques have attempted to neutralize adversarial triggers by leveraging limited training or in-distribution samples [9], [15], [16]. Another method adopts an input-level perspective by scaling pixel intensities of an image and checking consistency in the model's predictions [17]. Furthermore, researchers in [18] utilize a Vision Transformer (ViT) to classify previously seen adversarial attack patterns targeting traffic sign recognition systems in autonomous vehicles. These advances mark significant strides in safeguarding DNNs, but the persistence and evolution of adversarial threats demand continued innovation to stay ahead in this escalating arms race.

Contribution: Despite advancements in adversarial defense and training algorithms, achieving 100% protection against adversarial attacks remains elusive. These traditional methods are reactive, focusing on cleansing already-compromised models of embedded backdoor triggers. **Unlike the state-of-the-art methods, such as BDetCLIP [19], which utilize a poisoned model to uncover adversarial samples aimed at compromising the target model, our approach uses a clean and not poisoned CLIP model to detect unseen, open-world adversarial samples without having any prior knowledge about the attacks. This step is critical as millions of training samples are collected from the internet to train GenAI with little assurance that these samples are free from adversarial attacks.** This paper introduces a revolutionary and complementary strategy: a proactive algorithm designed to detect adversarial images before they wreak havoc. Our approach serves two critical purposes: *i) Pre-Training Defense:* Before training begins, the algorithm meticulously scans the dataset to identify and eliminate adversarial (backdoored) images that could poison object recognition models. This ensures the integrity and purity of the training data, safeguarding the foundation of model learning. *2) Inference-Time Shielding:* During inference, the algorithm acts as a vigilant gatekeeper, inspecting incoming images to block adversarial content from reaching the object recognition system. This prevents adversarial images from manipulating the model to misclassify inputs into the adversary's target class. By proactively identifying and neutralizing adversarial threats, this approach works in

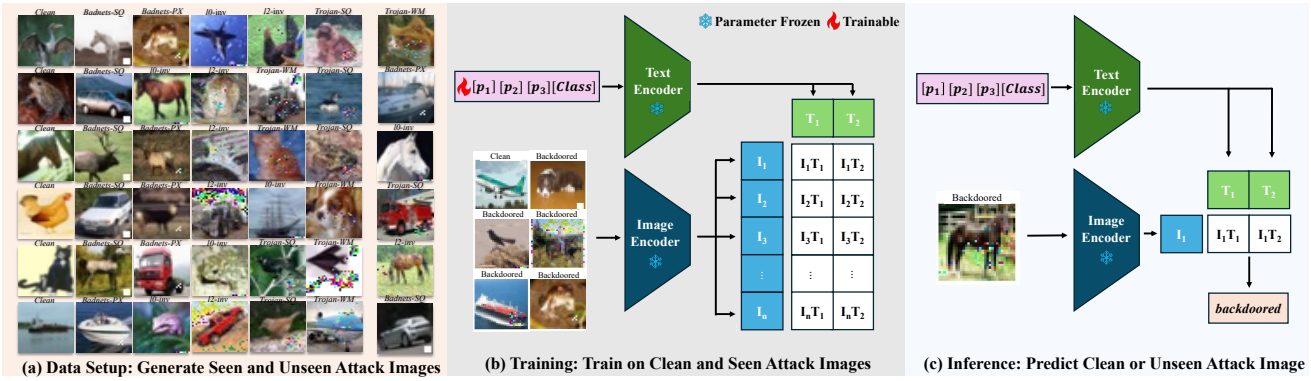

Fig. 1. The overall architecture of the proposed method on classifying unseen backdoor attack images.

tandem with traditional defense algorithms, creating a robust and comprehensive safeguard against adversarial attacks. It represents a significant leap forward in securing object recognition systems.

II. PROPOSED METHOD

II-A. Preliminaries and Insigints

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ represent a benign training set with N images, where $x_i \in \mathcal{X}$ is the i-th image and $y_i \in \mathcal{Y} = \{1, \ldots, K\}$ is its corresponding label, with K being the number of classes. If the goal of adversaries is to poison object recognition systems, they generate a poisoned dataset $\mathcal{D}_p$ to train the attacked model using either a standard loss function or an adversary-specified one. Specifically, $\mathcal{D}_p$ is composed of a modified subset of $\mathcal{D}$, denoted as $\mathcal{D}_m$, and a benign subset $\mathcal{D}_b$, such that $\mathcal{D}_p = \mathcal{D}_m \cup \mathcal{D}_b$, where $\mathcal{D}_b \subset \mathcal{D}$. The modified subset $\mathcal{D}_m$ is defined as $\mathcal{D}_m = \{(x', y') \mid x' = G_X(x), y' = G_Y(y), (x, y) \in \mathcal{D}_s\}$, where $\alpha = \frac{|\mathcal{D}_s|}{|\mathcal{D}|}$ is the poisoning rate, $\mathcal{D}_s$ is the subset of $\mathcal{D}$ selected to be modified, and G_X, G_Y are the adversary-specified generators for poisoned images and labels, respectively. For example, in BadNets [20], $G_X(x) = (1 - \beta) \odot x + \beta \odot t$, where $\beta \in \{0, 1\}^{C \times W \times H}$, where C, W and H are dimensions of input, t is the trigger pattern, and $\odot$ denotes the element-wise product. If the goal of advesaries is to attack object recognition systems during inference, given an 'unseen' image $\hat{x}$ with ground-truth label $\hat{y}$, the model will predict its poisoned version $G_X(\hat{x})$ as $G_Y(\hat{y})$. **It is important to note that the goal of this paper is not to train any model or defend against attacked models, but to detect backdoored images $G_X(x)$ and $G_X(\hat{x})$.**

VLM. To detect unseen backdoored images, we harness the extraordinary capabilities of VLMs, which excel at transforming input images into lower-dimensional, highly informative feature spaces. Models like Contrastive Language-Image Pre-training (CLIP) [21], trained on an immense dataset of image-text pairs (*i.e. 400 million samples*), are uniquely equipped to extract rich and versatile features from input images, regardless of their intended application. Their joint training with text data makes VLMs particularly well-suited for advanced techniques like prompt tuning. Prompt tuning has emerged as a groundbreaking method in leveraging VLMs for diverse applications. In this process, a learnable soft prompt is passed through the language encoder of the VLM while keeping the weights of both the image and language encoders frozen. This innovative approach tailors VLMs—originally trained in an unsupervised manner on broad image-text data—to excel in specific downstream tasks. VLMs have a word embedding module $E(\cdot)$ that converts the symbolic words to word embeddings, a text encoder $f_t(\cdot)$ that maps the word embeddings to the joint space, and an image encoder $f_I(\cdot)$ that maps input images to the joint space.

II-B. Backdoor Image Detection Framework

The framework of our proposed method for detecting unseen backdoored images is illustrated in Figure 1, showcasing a novel and powerful approach to tackling the complex challenge of detecting unseen backdoored attacks images. We leveraged prompt tuning of VLMs to transcend their general-purpose design to become specialized tools for detecting and classifying backdoored images with unparalleled precision and adaptability [25], [26], [27]. Figure 1(a) highlights the diverse permutations of backdoor attack images generated across the CIFAR-10 dataset, emphasizing the robustness of our method in handling a wide range of malicious image manipulations. Unlike traditional defenses focused on protecting against backdoored models, our goal is to directly detect backdoored images, combining training and test images into a unified dataset for this purpose.

In the training phase, depicted in Figure 1(b), we leverage the state-of-the-art architecture of CLIP, utilizing both a text encoder and an image encoder to project prompts and input images into a shared, semantically rich embedding space. Learnable soft prompts $[p_1, p_2, p_3]$ are prepended to the word embedding of the "class" label, which is ei-

Table I. Experimental Results of Unseen Attack Classification (Accuracy).

Dataset	Method	Trojan-WM	Trojan-SQ	l_2-inv	l_0-inv	Badnets-SQ	Badnets-PX	Average
CIFAR-10	Simple-CNN [22]	64.86 ± 7.14	75.10 ± 11.98	51.56 ± 0.41	49.93 ± 0.50	49.94 ± 0.42	50.10 ± 0.15	56.92 ± 3.43
	Deep-CNN [23]	76.37 ± 6.20	58.69 ± 6.72	55.77 ± 5.77	50.18 ± 0.34	50.14 ± 0.40	50.04 ± 0.06	56.87 ± 3.25
	ResNet-18 [24]	84.52 ± 5.22	75.40 ± 7.22	53.42 ± 1.28	51.17 ± 1.38	50.00 ± 0.00	54.29 ± 5.73	61.47 ± 3.47
	Proposed Method	**96.10 ± 0.38**	**96.44 ± 0.45**	**93.85 ± 0.69**	**96.45 ± 0.44**	**75.45 ± 2.03**	**58.89 ± 1.31**	**86.20 ± 0.88**
GTSRB	Simple-CNN [22]	82.86 ± 12.15	60.80 ± 10.35	53.58 ± 0.14	50.84 ± 0.30	50.00 ± 0.00	50.03 ± 0.03	58.02 ± 3.83
	Deep-CNN [23]	83.93 ± 14.91	61.61 ± 13.95	53.17 ± 0.20	51.45 ± 0.47	50.01 ± 0.20	50.04 ± 0.10	58.36 ± 4.97
	ResNet-18 [24]	74.40 ± 5.93	87.52 ± 6.81	52.75 ± 1.39	69.29 ± 14.85	50.01 ± 0.01	50.02 ± 0.05	64.00 ± 4.84
	Proposed Method	**94.89 ± 0.78**	**95.72 ± 0.69**	**94.41 ± 0.65**	**86.99 ± 0.62**	**85.03 ± 0.62**	**60.42 ± 1.19**	**86.24 ± 0.76**

Note: The values represent mean ± standard deviation over three random seeds. Bold indicates the best accuracy results for the unseen attack.

ther clean for normal images or backdoored for malicious ones. To speed up the convergence time, we initialize $[p_1, p_2, p_3]$ with the word embeddings of "a photo of", where $p_i \in \mathbb{R}^d$ and d is the dimension of the output of CLIP's word embedding $E(\cdot)$, and is equal to 512. This sequence, combining the learnable soft prompts and the word embedding of the "class," is passed through the text encoder of CLIP and normalized, producing embedding vectors $T_1 = f_t([p_1, p_2, p_3, E(\text{'clean'})])/\|f_t([p_1, p_2, p_3, E(\text{'clean'})])\|$ for clean images and $T_2 = \frac{f_t([p_1,p_2,p_3,E(\text{'backdoored'})])}{\|f_t([p_1,p_2,p_3,E(\text{'backdoored'})])\|}$ for backdoored images. These embeddings capture the contextual nuances of the respective image classes.

Simultaneously, the image encoder processes all clean and backdoored images to generate high-dimensional embeddings $(I_1, I_2, \ldots, I_n)$, where $I_i = \frac{f_I(x_i)}{\|f_I(x_i)\|}$. To optimize the model, similarity scores $(I_j \times T_1, I_j \times T_2)$ are computed between the jth image embedding and text embeddings T_1 and T_2, and a cross-entropy loss function is employed to fine-tune the system. This comprehensive framework not only ensures precise detection of seen backdoored images but also paves the way for identifying unseen backdoor attacks with exceptional accuracy and adaptability.

During inference, as shown in Fig. 1(c), the learned prefix embeddings $[p_1, p_2, p_3]$ are appended to the word embeddings of "clean" and "backdoored" and passed through the text encoder to generate the frozen text embeddings: $T_1 = f_t([p_1, p_2, p_3, E(\text{'clean'})])$, and $T_2 = f_t([p_1, p_2, p_3, E(\text{'backdoored'})])$. It is important to highlight that, although this process resembles the training phase, the prefix embeddings p_1, p_2, and p_3 are frozen during inference and remain unaltered. Meanwhile, the image encoder processes the input image x_j to compute its corresponding embedding I_j. Finally, the similarity scores between I_j and T_1, as well as I_j and T_2, are calculated and compared: Similarity(I_j, T_1), Similarity(I_j, T_2). These similarity scores determine whether x_j is classified as clean or backdoored. This architecture provides robust detection of unseen backdoored images by aligning embeddings of clean and adversarial images with their corresponding text embeddings in a shared multimodal space.

Training. During training, the model optimizes the alignment between the fixed visual embeddings, I_j, and the learnable text embeddings, $T_k, k \in \{1, 2\}$, to enable recognition of adversarial images. For each image x_j, the similarity scores for each class $c \in \{\text{"clean"}, \text{"backdoored"}\}$ are calculated using the scaled dot product $s_{j,k} = \alpha \times (I_j \cdot T_k)$, where α is a scaling factor that amplifies the logits, and $(\cdot)$ denotes the dot product operation. The scaling factor ensures that the logits are in a range suitable for the cross-entropy loss function.

The similarity scores $s_{j,k}$ are passed to the cross-entropy loss function, which encourages the model to assign higher similarity scores to the correct class. The cross-entropy loss function is defined as $L = -\frac{1}{N}\sum_{j=1}^{N} \log\left(\frac{\exp(s_{j,k})}{\exp(s_{j,1})+\exp(s_{j,2})}\right)$, where N is the batch size, k is the true class label for the j-th sample, and $s_{j,k}$ represents similarity scores for all classes k for sample j. The learnable prefix embeddings in the text encoder are optimized using the Adam optimizer [28].

Inference. During inference, for each input image x_j, we compute the similarity scores $s_{j,k}$ for each class k. However, instead of computing class probabilities using the softmax function, we select the class with the highest similarity score, resulting in the predicted class label $\hat{k}$ given by $\hat{k} = \arg\max_k s_{j,k}$. Notably, the model is tested on unseen attack images, leveraging the information learned from seen attacks during training to generalize to novel unseen images.

III. EXPERIMENTS

Attack Models. We have selected six renowned backdoor attacks to evaluate our proposed architecture: Badnets Square (Badnets-SQ) [20], Badnets Pixels (Badnets-PX) [20], Trojan Square (Trojan-SQ) [6], Trojan Watermark (Trojan-WM) [6], l_2-inv [29], and l_0-inv [29]. These attacks cover a wide range of backdoor conditions, including universality, label specificity, and variations in backdoor shape, size, and location.

Datasets. We conduct experiments using two datasets: CIFAR-10 [30] and GTSRB [23]. CIFAR-10 includes 50,000 training images and 10,000 test images across 10 classes. GTSRB consists of 39,209 training images and 12,630 test images of traffic signals, spanning 43 classes.

Experiment Settings. It is important to note that unlike adversarial attacks and defense literature that work on the model, we are working on images solely to detect backdoored images. Therefore, we do not train any model (e.g. ResNet-18) for our evaluation. We use CLIP's ViT-B/32, the smallest architecture in the CLIP family, chosen for

Table II. Cross-Generalization Results (Accuracy).

Unseen Attack	CIFAR-10 → GTSRB	GTSRB → CIFAR-10
Trojan-WM	76.54 ± 0.97	80.24 ± 0.83
Trojan-SQ	78.54 ± 1.09	81.78 ± 0.68
l_2-inv	78.81 ± 1.59	74.95 ± 2.32
l_0-inv	73.68 ± 1.02	76.20 ± 1.95
Badnets-SQ	70.85 ± 0.28	75.69 ± 0.14
Badnets-PX	62.75 ± 0.51	62.84 ± 0.26

Note: The values represent mean ± standard deviation over three random seeds. *Train* → *Test*.

its efficiency in balancing performance and computational demands. We leverage the predefined training and testing splits from the previously mentioned datasets. Clean images are taken directly from the train and test split without modification, while backdoored images are generated by applying the predefined attack types to all clean images. To evaluate the performance of our proposed method in detecting **unseen attacks**, only five of six attack types are selected for training. To maintain balance, we randomly select an equal number of images from each attack type to match the total number of clean images. At inference, the model is tested on all clean test images and their backdoored version by the **unseen attack**, which is the attack type excluded during training. Furthermore, we set the scaling factor α = 100 and the Adam optimizer's learning rate for the learnable prefix embeddings to 10^{-5}. Training is conducted over 10 epochs with a batch size of 128.

III-A. Experimental Results

To compare the performance of our methods in detecting unseen backdoored images, we train three widely used CNN architectures: Simple-CNN [22] consists of three convolutional layers and a fully connected, Deep-CNN [23] consists of six convolutional layers with a dropout layer and fully connected layer for classification, and ResNet-18 [24].

Table I presents the evaluation results, showcasing the exceptional performance of our proposed method in classifying unseen backdoor attack images. For example, on the CIFAR-10 dataset, our method achieves detection accuracies exceeding 95% for Trojan-WM, Trojan-SQ, and *l0*-inv triggers, alongside an impressive 93.85% accuracy for *l2*-inv triggers. Furthermore, the results highlight a 25% increase for detecting Badnets-SQ triggers and over 4.5% on Badnets-PX. To further validate the robustness of our approach, we extend our experiments to the GTSRB dataset. While prior CNN-based methods show promise in detecting Trojan triggers, our proposed method exhibits a remarkable performance increase, achieving an accuracy improvement of approximately 11% and 8% on Trojan-WM and Trojan-SQ, respectively. Additionally, our method significantly enhances detection accuracy by 35% for unseen Badnets-SQ triggers on the GTSRB dataset. While our method performs well across most unseen attack types, the lower accuracy on Badnets-PX attacks highlights a limitation in detecting subtle, pixel-level triggers. The minimal changes may not significantly alter the global image features captured by the frozen visual encoder, making them more challenging to detect. Overall, these results validate the value of our approach in detecting unseen backdoor attacks across both datasets.

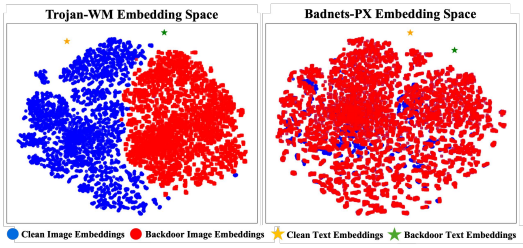

Fig. 2. t-SNE Visualization of test embeddings of Trojan-WM and Badnets-PX attacks on the CIFAR-10 dataset.

III-B. Cross-Generalization Experiment

Ensuring robust generalization across datasets is crucial for backdoored image detection, particularly when facing unseen triggers. To evaluate the strength of our proposed approach regardless of the dataset used during the training, we conduct experiments to train on one dataset (e.g. CIFAR-10) and test on the other (e.g. GTSRB), while ensuring that the model is still trained on seen triggers and tested on unseen triggers. Table II illustrates the impressive results of these experiments. For instance, in the initial tests (CIFAR-10 → GTSRB), our model achieves an average accuracy of 77.54% on unseen Trojan triggers. The model retains robust performance across unseen *l2*-inv and *l0*-inv and triggers, achieving accuracies of 78.81% and 73.68%, respectively.

When reversing the train and test sets (GTSRB → CIFAR-10), Trojan triggers achieve a higher average accuracy of 81.01%. In this scenario, the model once again performs well in identifying unseen *l2*-inv and *l0*-inv and triggers. Interestingly, the model detects 62.84% of unseen Badnets-PX triggers, which outperforms the performance when training directly on CIFAR-10. This improvement likely occurs due to the greater diversity and visual complexity of the GTSRB dataset (*i.e.* 43 classes) compared to less diverse CIFAR-10 dataset (*i.e.* 10 classes). This diversity appears to enable the model to learn more generalized representations, improving its ability to detect subtle pixel-level triggers like Badnets-PX.

III-C. Visual Analysis

To demonstrate the separation between clean and adversarial embeddings, we present t-SNE visualizations [31] of the test image and text embeddings within the embedding space for unseen backdoor triggers Trojan-WM and Badnets-PX, shown in Figure 2. These attacks were selected for illustration due to their contrasting detection performances:

Table III. Learnable vs. Static Prefix (Accuracy).

Unseen Attack	CIFAR-10		GTSRB	
	[p1][p2][p3]	"a photo of"	[p1][p2][p3]	"a photo of"
Trojan-WM	96.10 (+42.87)	53.23	94.89 (+22.28)	72.61
Trojan-SQ	96.44 (+43.14)	53.30	95.72 (+28.46)	67.26
l_2-inv	93.85 (+39.84)	54.01	94.41 (+20.30)	74.11
l_0-inv	96.45 (+44.06)	52.39	86.99 (+22.41)	64.58
Badnets-SQ	75.45 (+24.14)	51.31	85.03 (+29.90)	55.13
Badnets-PX	58.89 (+8.99)	49.90	60.42 (+8.83)	51.59

Note: Differences in parentheses represent the accuracy improvement of the learned prefix over the static prefix. Values are shown in blue.

Trojan-WM achieves over 96% accuracy, while Badnets-PX achieves around 59%. For Trojan-WM, the correct text embeddings are closely aligned with their corresponding image embedding clusters, helping in create a distinct separation between clean and backdoor images. In contrast, the embedding space for Badnets-PX reveals a less distinct clustering pattern. While some separation occurs between the text embeddings, there is a significant overlap in image embeddings. This misalignment makes it more challenging to distinguish the unseen Backdoor-PX images from clean images during inference.

III-D. Ablation on Fixed Prefix

While our approach is built on leveraging a learnable prefix to adapt the textual representation in detecting unseen backdoored images, it is important to examine the effect on the model if the prefix remains static. We compare the results when using a fixed prompt – "a photo of" – with the performance using our learned prefix, shown in Table III. When the prompt remains static, the model relies heavily on the understanding of the fixed prefix, providing no additional semantic context that highlights backdoored features. Furthermore, the base model is applying only pre-trained knowledge without any fine-tuning, leading to a lack of generalization to previously unseen images of backdoored images. However, the learnable prefix helps guide the model's attention by enabling the prompt to dynamically adapt to associated backdoor patterns. This helps align visual and textual embeddings in the multimodal embedding space, making it more effective at detecting unseen backdoor triggers.

IV. CONCLUSION

Defending object recognition systems against adversarial attacks has traditionally centered on reactive strategies like cleansing backdoored models or adversarially training them. In this groundbreaking work, we introduced a paradigm shift: a proactive method for detecting unseen backdoored (poisoned) images before they can infiltrate object recognition systems. Our approach serves dual purposes—vetting training datasets to ensure integrity and safeguarding inference by blocking adversarial images before they reach the model. We achieved this by harnessing the unparalleled generalization capabilities of vision-language models like CLIP, leveraging prompt tuning to exploit their training on vast and diverse datasets. Extensive experiments across six distinct types of unseen attacks demonstrate the robustness and effectiveness of our approach, setting a new benchmark for proactive defense mechanisms. While this pioneering work represents the first step toward detecting backdoored images, future research must delve deeper into improving detection of pixel-based attacks, where subtle, localized triggers present a formidable challenge. This study paves the way for a new era in securing object recognition systems against adversarial threats.

V. REFERENCES

[1] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4804–4814.

[2] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Elasticface: Elastic margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1578–1587.

[3] A. Mahyari, "Policy augmentation: An exploration strategy for faster convergence of deep reinforcement learning algorithms," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3505–3509.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2019.

[5] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 182–199.

[6] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[7] L. Song, X. Yu, H.-T. Peng, and K. Narasimhan, "Universal adversarial attacks with natural triggers for text classification," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2021.

[8] Y. K. Singla, S. Parekh, S. Singh, C. Chen, B. Krishnamurthy, and R. R. Shah, "Minimal: mining models for universal adversarial triggers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 330–11 339.

[9] Y. Shi, M. Du, X. Wu, Z. Guan, J. Sun, and N. Liu, "Black-box backdoor defense via zero-shot image pu-

rification," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[10] D. Yuan, S. Wei, M. Zhang, L. Liu, and B. Wu, "Activation gradient based poisoned sample detection against backdoor attacks," *arXiv preprint arXiv:2312.06230*, 2023.

[11] S. Wei, M. Zhang, H. Zha, and B. Wu, "Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] X. Chen, W. Wang, C. Bender, Y. Ding, R. Jia, B. Li, and D. Song, "Refit: a unified watermark removal framework for deep learning systems with limited data," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021, pp. 321–335.

[13] M. Weber, X. Xu, B. Karlaš, C. Zhang, and B. Li, "Rab: Provable robustness against backdoor attacks," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1311–1328.

[14] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2021.

[15] S. Chai and J. Chen, "One-shot neural backdoor erasing via adversarial weight masking," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 285–22 299, 2022.

[16] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[17] J. Guo, Y. Li, X. Chen, H. Guo, L. Sun, and C. Liu, "Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency," *arXiv preprint arXiv:2302.03251*, 2023.

[18] F. Mumcu and Y. Yilmaz, "Fast and lightweight vision-language model for adversarial traffic sign detection," *Electronics*, vol. 13, no. 11, p. 2172, 2024.

[19] Y. Niu, S. He, Q. Wei, Z. Wu, F. Liu, and L. Feng, "Bdetclip: Multimodal prompting contrastive test-time backdoor detection," *arXiv preprint arXiv:2405.15269*, 2024.

[20] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[22] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1625–1634.

[23] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, and K. Keutzer, "How much can clip benefit vision-and-language tasks?" *arXiv preprint arXiv:2107.06383*, 2021.

[26] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[27] ——, "Conditional prompt learning for vision-language models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 816–16 825.

[28] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2020.

[30] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[31] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.