This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

Privacy Preserving Defense For Black Box Classifiers Against On-Line Adversarial Attacks

Rajkumar Theagarajan, *Student Member, IEEE,* and Bir Bhanu, *Life Fellow, IEEE*

Abstract—Deep learning models have been shown to be vulnerable to adversarial attacks. Adversarial attacks are imperceptible perturbations added to an image such that the deep learning model misclassifies the image with a high confidence. Existing adversarial defenses validate their performance using only the classification accuracy. However, classification accuracy by itself is not a reliable metric to determine if the resulting image is "adversarial-free". This is a foundational problem for online image recognition applications where the ground-truth of the incoming image is not known and hence we cannot compute the accuracy of the classifier or validate if the image is "adversarial-free" or not. This paper proposes a novel privacy preserving framework for defending Black box classifiers from adversarial attacks using an ensemble of iterative adversarial image purifiers whose performance is continuously validated in a loop using Bayesian uncertainties. The proposed approach can convert a single-step black box adversarial defense into an iterative defense and proposes three novel privacy preserving Knowledge Distillation *(KD)* approaches that use prior meta-information from various datasets to mimic the performance of the Black box classifier. Additionally, this paper proves the existence of an optimal distribution for the purified images that can reach a theoretical lower bound, beyond which the image can no longer be purified. Experimental results on six public benchmark datasets namely: 1) Fashion-MNIST, 2) CIFAR-10, 3) GTSRB, 4) MIO-TCD, 5) Tiny-ImageNet, and 6) MS-Celeb show that the proposed approach can consistently detect adversarial examples and purify or reject them against a variety of adversarial attacks.

Index Terms—Adversarial Defense, Bayesian uncertainties, Black box defense, Ensemble of defenses, Image purifiers, Knowledge distillation, Privacy preserving defense.

1 Introduction

Although deep learning has had astounding success on several image classification tasks, it has been shown to be vulnerable to adversarial attacks [1] - [3]. Adversarial attacks to an image are carefully crafted perturbations that are so subtle that a human observer does not even notice the modification at all, but can cause deep learning models to misclassify the input. Adversarial attacks can broadly classified into two categories namely: 1) white box, and 2) black box attacks. White box adversarial attacks assume that the adversary has partial/full knowledge about the target model's architecture, parameters, and training data [4], whereas in the black box attack, the adversary lacks this information [5]. In this paper we focus on defending against black box adversarial attacks.

Current defenses against adversarial attacks can be classified into four approaches: 1) modifying the training data, 2) modifying the model, 3) using auxiliary tools, and 4) detecting and rejecting adversarial examples. Modifying the training data involves augmenting the training dataset with adversarial examples and re-training the classifier [6] - [8] or performing N number of pre-selected image transformations in a random order [9] - [12]. Modifying the model involves pruning the architecture of the classifier [14] - [16] or adding pre/post-processing layers to it [21] - [23]. The category 1) and 2) approaches are not compatible to be used for defending Black box models because they require information about the target model's architecture/training data which are not known.

Using auxiliary tools involves having an independent module that is able to process the input before it is passed to the classifier [26] - [28]. Detecting and rejecting adversarial examples involve using domain adaptation techniques and carrying out statistical analysis [30] - [33] to detect adversarial examples. The approaches in the latter two categories are the most suitable for defending Black box classifiers as they do not require any information about the target model's architecture/training data. However, a drawback of these approaches is that they do not quantify how much adversarial component is left in the resulting purified image. Further, these approaches are single-step defenses which means that the defense can purify the image only once and the defense assumes that the purified image is void of all adversarial components. These defenses are trained on annotated datasets and after achieving a reasonable performance, they are deployed to defend real-world applications. Once deployed it is assumed that all of the purified images are no longer adversarial, but in reality this is not the case.

A foundational problem for online and safety-critical applications [35] - [39] is that it is not possible to know the annotations of all of the incoming input images, and these single-step defenses do not have the ability to determine by themselves whether the purified image is adversarial or not which could further cause disastrous results. To address this problem, this paper proposes a general framework for defending black box classifiers from adversarial attacks using an ensemble of iterative

- *This work was supported in part by NSF grant 1911197 and Bourns endowment funds. The contents of the information do not reflect the position or the policy of the US Government*
- *Rajkumar Theagarajan and Bir Bhanu are with the Center for Research in Intelligent Systems, University of California, Riverside, CA 92521 USA (e-mail: rthea001@ucr.edu and bhanu@vislab.ucr.edu).*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 1: Summary of related work for white box adversarial defense

Approach	Authors	Methods and Comments
Adversarial training	Goodfellow *et al*. [3], Huang *et al*. [6] Tramèr *et al*. [7], Zhang *et al*. [8]	Adversarial examples are introduced into the dataset to improve the robustness of the training model with the legalized adversarial examples.
Gradient hiding	Tramèr *et al*. [7]	Hides the information about the gradient from adversaries. However, learning a *Substitute* model through knowledge distillation can penetrate this defense.
Data compression	Das *et al*. [9], Dziugaite *et al*. [10] Xu *et al*. [23]	Used JPEG compression on input images and color scale normalization as a pre-processing step to defend classifiers.
Data randomization	Xie *et al*. [24], Wang *et al*. [25]	Performed random resizing and addition of random noise to reduce the effect of the adversarial perturbations.
Image transformation	Guo *et al*. [11], Raff *et al*. [12]	Used hand-crafted image transformations in a random order to remove the effect of adversarial perturbations.
Blocking the transferability	Hosseini *et al*. [13]	Performed three step null labeling method that classifies adversarial examples into a ”null” class.
Defensive distillation	Papernot *et al*. [15]	Defensive distillation is done by converting the softmax output logit probabilities to create soft labels.
Thermometer encoding	Buckman *et al*. [21]	Discretized the input by replacing pixel values with thermometer encoding.
Feature denoising	Liao *et al*. [14], Xie *et al*. [16] Mustafa *et al*. [17]	Used feature disentanglement to defend against adversarial attacks.
Architecture pruning	Liu *et al*. [18], Dhillon *et al*. [19] Ye *et al*. [20]	Reduced the effect of the adversarial perturbation by pruning the architecture and weights of the original classifier.

TABLE 2: Summary of related work for black box adversarial defense

Approach	Authors	Methods and Comments
Defense-GAN	Samangouei *et al*. [101]	Used Generative Adversarial Networks (GAN) to minimize the reconstruction error.
PixelDefend	Song *et al*. [27]	Used a PixelCNN to purify adversarial examples.
ShieldNets	Theagarajan *et al*. [28]	Used Probabilistic Adversarial Robustness (PAR).
MagNet	Meng *et al*. [26]	Used the reconstruction loss of an autoencoder to detect and purify adversarial attack.
SafetyNet	Lu *et al*. [29]	Detected if an image is adversarial using RGBD images.
Statistical testing	Grosse *et al*. [31]	Used kernel based two sample test to distinguish adversarial examples from the dataset.
Policy based detection	Lin *et al*. [32]	Compared action distributions from previous replays to detect policy targeting attacks.
Bayesian uncertainty	Rawat *et al*. [73] Gal and Ghahramani [74]	Distinguished between adversarial and original images using Bayesian uncertainties.
Ensemble of iterative adversarial defenses	Theagarajan and Bhanu [94]	Used an ensemble of adversarial defenses to defend black box facial recognition classifiers against adversarial attacks.
	Theagarajan and Bhanu (This paper)	**1) Proposed a novel model agnostic defense framework that uses an ensemble of adversarial defenses to iteratively purify adversarial images for various black box applications, 2) showed the relationship between the adversarial image and its corresponding purified image and, 3) proved the existence of a theoretical lower bound in the input space beyond which the image cannot be further purified.**

adversarial defenses whose performance is continuously validated in a loop using Bayesian uncertainties. The paper proposes three novel knowledge distillation approaches for transferring the functionality of the black box classifier into our defense. The experimental results on six different datasets shows that our defense can be applied to defend various black box applications ranging from the general Fashion-MNIST [78] and CIFAR-10 [79] datasets to face biometrics and classification of vehicles and traffic signs for autonomous driving. In summary, the contributions of this paper are as follows:

• To the best of our knowledge, this is the first approach that defends against adversarial attacks using an ensemble of iterative adversarial defenses and can convert any single-step defense into an iterative adversarial defense.
• We prove theoretically and demonstrate empirically that there exists a lower bound in the input space on the amount of purification carried out on an image beyond which it can no longer be purified.
• The paper proposes three novel privacy preserving knowledge distillation approaches that exploit prior meta-information of the training dataset in order to transfer the functionality of the black box classifier into our defense and it does not require any information such as the logits probabilities [55], [56] or *Teacher model* architecture [63], [64].
• To the best of our knowledge there is no other work that defends black box models from adversarial attacks without any information of the model's trained parameters or the dataset that are used for training the model prior to our work in [94].
• Exhaustive evaluation on six public benchmark datasets shows that are approach is able to consistently purify/reject adversarial examples. Various ablation studies show that it is computationally expensive to break the defense of our framework compared to stand-alone defenses.

2 Related Work

In this section we describe state-of-the-art black box adversarial defenses and knowledge distillation approaches and contrast them with our approach. Tables 1 - 3 show a summary of the related work done for white box, black box adversarial defense, and knowledge distillation, respectively. In contrast to the related work, our work is significantly different in the following aspects:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 3: Summary of the related work for knowledge distillation

Authors	Comments
Orekondy *et al.* [56]	Used the softmax logit probability scores and cross dataset images to distill knowledge.
Furlanello *et al.* [55]	Decomposed the predictions of the white box model into incorrect prediction and ground-truth information.
Frosst and Hinton [63]	Distilled deep networks to decision trees in order to explain the predictions.
Ba and Caruana [64]	Distilled knowledge using the L_2 norm between the logits of the white box Teacher and Student model.
Shin *et al.* [65]	Used knowledge distillation for the purpose of minimizing forgetting in continuous learning
Tan *et al.* [66]	Performed distillation using the difference in prediction between a white box *Student* and *Teacher* classifier.
Wang *et al.* [67]	Explained the behavior of white box deep neural networks by transferring the functionality to a smaller model.
Song *et al.* [68]	Used a dual branch student neural network to learn the latent space which is guided by a white box teacher using Attentive Knowledge Distillation (AKD).
Tessler *et al.* [69]	Proposed a hierarchical learning framework that consists of multiple Deep Skill Networks (DSN) to perform knowledge transfer and life long learning.
Crowley *et al.* [70]	Proposed structural model distillation to reduce the memory cost using block convolutions that produces a student architecture that is a simple transformation of the white box teacher architecture.
Theagarajan and Bhanu [94]	Used publicly available crowd sourced images to probe and distill the knowledge of a black box classifier.
Theagarajan and Bhanu (This paper)	**Proposed three knowledge distillation approaches that exploit the prior meta-information of the training datasets and showed that knowledge distillation can still be performed even when we do not have any knowledge such as the logit probabilities [58], [59] or architecture about the black box classifier [63], [64]**

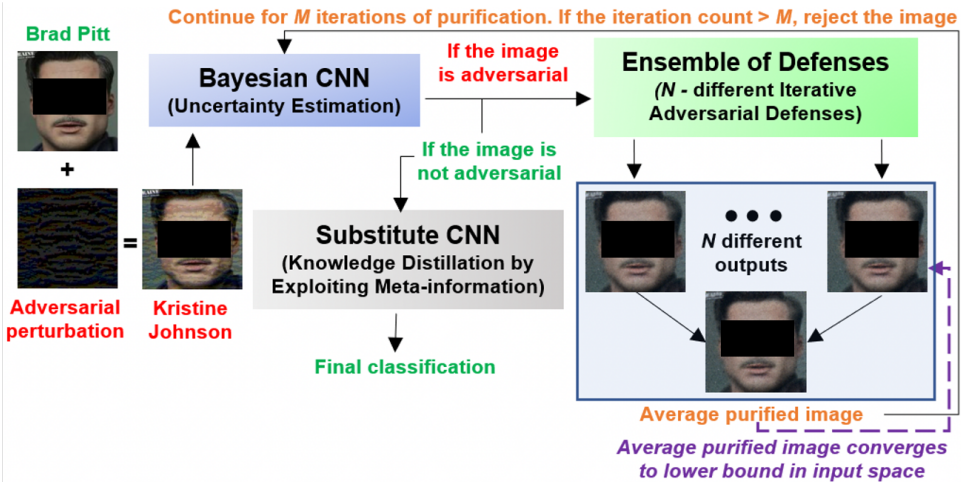


Fig. 1: Overall framework of our approach. Faces are masked to hide the identity of individuals.

• **Ensemble of Iterative Adversarial Attacks**: To the best of our knowledge, there is no other work that can convert single-step adversarial defenses into an iterative defense and also quantify the amount of adversarial component in the output of the defense after each iteration.
• **Relationship between Adversarial and Non-adversarial images**: This paper proves the existence of a lower bound in the input space beyond which an image can no longer be purified. We empirically verify this by showing the convergence of all the individual defenses across multiple iterations of purification.
• **Knowledge Distillation using Meta-information**: This is the first approach that exploits only the meta-information of a dataset to perform knowledge distillation on strict black box classifiers. Unlike the related work shown in Table 6 we assume we know neither the architecture of the classifier, the output logit probabilities, nor the data distribution used for training.

3 Technical Approach

Fig. 1 shows the overall framework of our approach. The input image (X) first passes through the Bayesian Convolutional Neural Network (CNN) and if the image is classified as an adversarial image, it is purified by the ensemble of independently trained iterative adversarial defenses. The purified image from each defense algorithm is averaged resulting in the average purified image (X'_i), where i refers to the current iteration number. Next, X'_i is passed as input back to the Bayesian CNN. If X'_i is not adversarial, it is passed as input to the Black box classifier for final classification, else, it is passed again as input to the ensemble of adversarial defenses and this continues for M iterations. After M iterations, if X'_M is still adversarial then the image is rejected.

In Fig. 1, we chose to take the average of the purified image of each individual defense as this further helps in removing the high frequency adversarial perturbations [40], [41] that may have not been purified by some of the individual defenses. Additionally, the number of iterations for purification, M, is chosen empirically depending on the training dataset used (see Section 3.4.1) and we prove the existence of a theoretical lower bound beyond which an image cannot be further purified (see Section 3.4.2).

3.1 Assumptions and Target Applications of our Defense

• ***Assumptions of our Defense***: **1)** The output of the Black box model is the top predicted class without any probabilities.

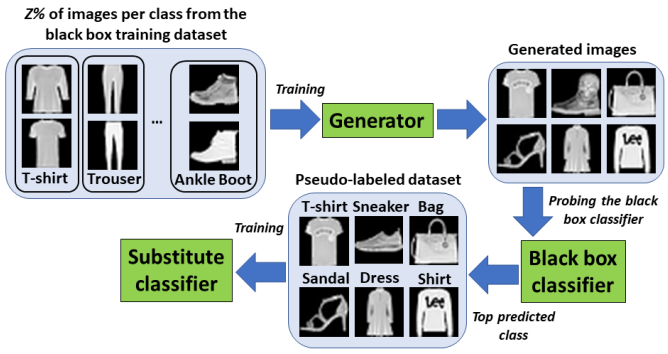
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

2) The architecture, parameters and *entire* training dataset of the Black box are not known to both the adversary and the defense algorithm. **3)** The outputs of the Bayesian framework and ensemble of adversarial image purifiers are not shared with the adversary (i.e., the adversary can only see the final classification of the Black box). Although this is a strong assumption, we perform ablation studies to estimate the computational burden required to break the defense, when the adversary knows partial information about our defense strategy (see Section 4.6).

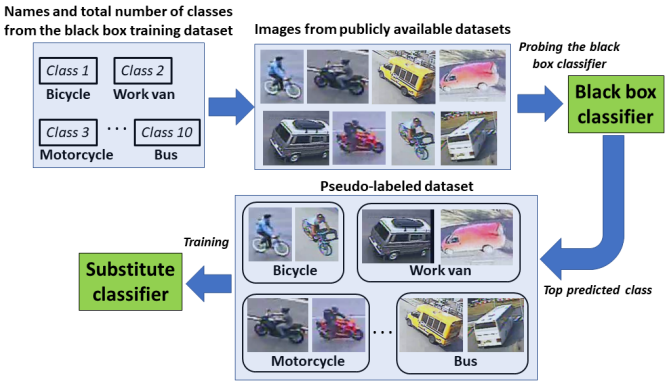
• ***Reasons for our Assumptions***: When black box classifiers are made publicly available, they must neither reveal the sensitive details of their training data nor the features encoded within the classifier, but they must still be accurate and solve the underlying learning task. This kind of protection is also known as *Security through Obscurity* [42] and it is important that these models preserve the privacy of their sensitive training data. Advances in the field of *Differential Privacy Learning* have shown that it is possible to train black box classifiers without compromising the privacy and personnel information contained within the training dataset [43] - [46]. On the contrary, recent works have also shown that it is possible for an adversary to create adversarial attacks to determine if a specific data point was used to train the model and thus causing a leak in sensitive information [47] - [50]. There have also been similar news articles such as the ban on facial recognition technology in the state of California, USA [51] due to the misuse of information and privacy concerns. Once personally identifiable information is leaked, it is impossible to reverse it, hence it is imperative that adversarial defenses protect black box classifiers against adversarial attacks while at the same time maintain privacy. For these reasons we assume that our defense has little to no knowledge of the training data as well as the information on the learned features of the black box classifier. Our proposed defense approach is suited for black box applications that require security against adversarial attacks and preserve user privacy without giving away any sensitive information. These include a wide variety of machine intelligence applications such as human biometrics [94], [115], remote surveillance [52], autonomous driving [53], etc.

3.2 Functionality Transfer via Knowledge Distillation

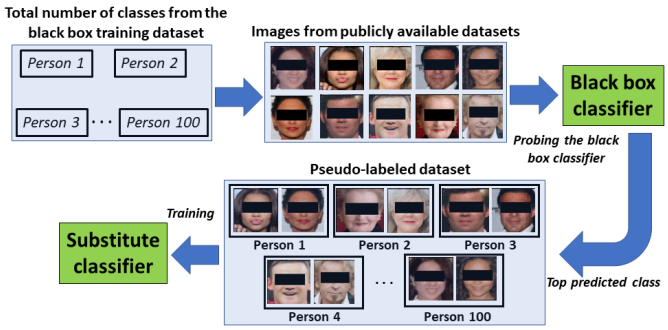
According to the assumptions described in Section 3.1, we do not have any information about the target black box model's architecture, parameters, and training dataset, hence we cannot directly use our defense algorithms. To overcome this problem, we transfer the functionality of the black box model to a substitute model using knowledge distillation [54] - [56]. In this paper we assume that the target black box classifier is deterministic (weights are fixed after training) and ignore approaches that continuously update the weights of the CNN (e.g., by using incremental [57] and reinforcement learning [58]). Based on the above discussion, we propose three novel Knowledge Distillation (KD) approaches namely: KD-1, KD-2, and KD-3 that exploit the meta-information related to the training datasets. Unlike the approaches proposed in Table 3, our knowledge distillation method is different because the approaches in Table 3 assume that they have prior knowledge about the training



(a) Illustration of KD-1



(b) Illustration of KD-2



(c) Illustration of KD-3

Fig. 2: Illustration of (a) KD-1 using the Fashion MNIST dataset, (b) KD-2 using the MIO-TCD dataset, and (c) KD-3 using the MS-Celeb dataset.

dataset or the logit probability of outputs and/or architecture of the target black box classifier. In our approach, we assume a strict black box classifier where we have no knowledge about the architecture or logit probabilities of the black box classifier and the only observable output is the single top predicted class. Fig. 2 shows illustrations of our three knowledge distillation approaches.

Assumptions of our KD approaches:

• ***KD-1***: We know the total number of classes, the names of all the classes in the training dataset and Z% of images from each class.

• ***KD-2***: We know only the total number of classes, the names of all the classes in the training dataset and the domain the dataset belongs to.

• ***KD-3***: We know only the total number of classes in the training dataset and the domain the dataset belongs to.

In KD-2 and KD-3, domain refers to the application of the black box classifier, e.g., classifiers trained on the MS-

Celeb [59] and MIO-TCD [60] datasets generally belong to the domain of face recognition and vehicle classification, respectively.

3.2.1 Knowledge Distillation-1 (KD-1)

For this case we assume that we have access to Z% of images of each class from the training dataset used to train the black box classifier as shown in Fig. 2(a). For simplicity we refer to the black box classifier as the *Teacher model* and our substitute classifier as the *Student model*. Next, we randomly select Z% of images from each class of the dataset and use these images to train a Deep Convolutional Generative Adversarial Network (DCGAN) [72]. After training the DCGAN, we probe the *Teacher model* with the images generated using DCGAN and label them as the predicted class. We then augment these labeled images to the Z% amount of original images to create a pseudo-labeled dataset. In order to have an equal data distribution between the original and pseudo-labeled dataset, we made the number of images per class in the pseudo-labeled dataset to be the same as the original dataset used for training the *Teacher model*. Finally, the pseudo-labeled dataset is used for training the *Student model* (i.e., our substitute classifier).

In KD-1, since we randomly select Z% of images from each class to train the DCGAN and create the pseudo-labeled dataset, there is no guarantee that the selected images will statistically represent an accurate distribution of each class. To address this issue we perform *k*-fold cross validation by selecting different Z% amount of images (see Section 4.5). In our approach we set the value of Z to be 25% and 50% and use the Fashion-MNIST [78], CIFAR-10 [79], and GTSRB [80] datasets to evaluate KD-1.

3.2.2 Knowledge Distillation-2 (KD-2)

In KD-2, we do not have any knowledge about the images or their data distribution, but we know the names of all the classes and the total number of classes in the dataset as shown in Fig. 2(b). Hence, we search for images belonging to those classes from publicly available datasets. First, we create a pseudo-labeled dataset by probing the *Teacher model* with the images from the public domain and label these images with the predicted class. It should be noted that since we already know the names of the class the image belongs to, we do not need to further probe the *Teacher model* and re-label the images, but prior work done by [81] - [83] shows that when we train the *Student model* with images that were manually annotated by humans, the classification accuracy with respect to the *Teacher model* is lower compared to training the *Student model* with images that were annotated entirely by the *Teacher model*. The reason for this is that images that are misclassified by the *Teacher model* add a regularizing effect while training the *Student model*, thus, resulting in efficient functionality transfer [83].

Finally, after creating the pseudo-labeled dataset, we use it for training the *Student model*. In this paper we use the MIO-TCD [60] and Tiny ImageNet [84] dataset to evaluate the KD-2 approach.

3.2.3 Knowledge Distillation-3 (KD-3)

In this setting we know only the total number of classes in the training dataset and the domain of the dataset as shown in Fig. 2(c). We neither have any knowledge of the images in the training dataset nor we know the names of the classes. This scenario occurs in large scale identification applications such as face recognition and pedestrian re-identification. Based on this we scour for images that belong to this domain from publicly available datasets. Similar, to KD-2 we probe the black box classifier with these images in order to create the pseudo-labeled dataset and then use the pseudo-labeled dataset to train the substitute classifier. We evaluate the KD-3 approach using the MS-Celeb dataset [59] that consists of facial images of over 99,000 celebrities. It should be noted that in this paper we evaluate the KD-2 and KD-3 approaches in two different settings: i) when there is no overlap and ii) when there is 50% overlap between the black box training dataset and the pseudo-labeled dataset (see Table 6).

Comments: In both KD-1 and KD-2, although the *Student model* is trained on a dataset that is entirely different from the dataset used for training the *Teacher model*, we are still able to distill some of the learned features from the *Teacher model*. The reason for this is that the *Teacher model* is assumed to be a deterministic model meaning that, after training, the features learned are fixed and do not change over time. Hence, when we probe a given image X, the resulting prediction $f(X) = Y$ will never change and with a considerably large and diverse pseudo-labeled dataset, the *student model* is able to distill the learned features of the *Teacher model* and achieve good classification accuracy. This observation is particularly advantageous to black box applications that have an abundance of crowd-sourced images from various publicly available datasets such as the field of face biometrics and vehicle classification. *Note that prior to [94], this observation had not been addressed* and can effectively be used for distilling the knowledge of robust Black box classifiers into edge devices that usually have compressed and shallow CNN architectures [95] - [98].

3.3 Uncertainty Prediction via Bayesian Learning

In the domain of adversarial defense, it is very important to know the amount of adversarial perturbation that still remains in the output image after it passes through any defense algorithm. Bayesian methods offer a principled way to represent these uncertainties in a model and can be utilized to quantify a model's confidence in its prediction [73]. Deep learning models $f(\cdot)$ consists of a set of weights w that are optimized on a labeled dataset $D = \{x_i, y_i\}_{i=1}^{N}$, where x_i and y_i are the input data and corresponding ground-truth, respectively. Bayesian inference involves learning a posterior distribution over the weights $p(w|D)$ which is used for predicting unseen observations:

$$p(y|x, D) = \int p(y|x, w)\, p(w|D) dw \tag{1}$$

The above integral is intractable because of the sheer number of parameters in deep learning models. To overcome this, in our approach we design the Bayesian CNN using *Bayes by Backprop* [75]. *Bayes by Backprop* is a variational

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

inference to Bayesian neural networks where the posterior is assumed to be a diagonal Gaussian distribution which assumes independence among the variables. The Gaussian posterior $q_\theta(\omega|D)$ is defined to be as similar as possible to the original posterior $p(\omega|D)$ when measured by the KL divergence [76]. Based on this the optimal parameters are defined as:

$$\theta_{opt} = \arg\min_{\theta} KL(q_\theta(w|D)||p(w)) - \mathbb{E}_{q(w|\theta)}(log\ p(D|w)) + log\ p(D) \quad (2)$$

After learning the approximate posterior distribution we compute two uncertainty metrics namely: 1) Aleatoric, and 2) Epistemic uncertainties [77] which are given by:

$$Aleatoric_{\ Uncertainty} = \frac{1}{S}\sum_{i=1}^{S} diag(\hat{g}_i) - \hat{g}_i\,\hat{g}_i^T \quad (3)$$

$$Epistemic_{\ Uncertainty} = \frac{1}{S}\sum_{i=1}^{S}(\hat{g}_i - \tilde{g})(\hat{g}_i - \tilde{g})^T \quad (4)$$

where, S is the number of samples drawn from the posterior distribution, $\tilde{g} = \frac{1}{S}\sum_{i=1}^{S}\hat{g}_i$ and $\hat{g}_i = f_{w_i}(x)$. It should be noted that we trained the Bayesian CNN using the same pseudo-labeled dataset used for training the *Substitute* model as described in Section 4.4 and Table 6.

Aleatoric Uncertainty: It is a measure for the variation of data. This value increases if certain classes are heavily unbalanced or lack data. Adversarial images have been shown to lie in the high frequency and low probability density regions [27], [28], which is similar to highly unbalanced and long-tailed datasets. This is also a reason why adversarial training [3] is an effective white box defense.

Epistemic Uncertainty: It is caused by the model itself. It is the ability of a model to learn robust and representative features which depends on its architecture and parameters. This value increases in the presence of adversarial attacks.

3.3.1 When is an image adversarial?

After learning the posterior distribution $q_\theta(\omega|D)$, we find the minimum uncertainty for an adversarial image. For this we generated adversarial images with the smallest perturbation (i.e. $\epsilon = 1/255$) for the *Substitute* model using three well known attacks: IFGSM [3], BIM [34], and PGD [100]. Next, these adversarial images are transferred to the Bayesian CNN and we compute the average (μ) and standard deviation (σ) of the Epistemic and Aleatoric uncertainties. Finally, we set two thresholds T_1 and T_2 given by:

$$T_1 = \mu(Aleatoric) - 3\sigma(Aleatoric) \quad (5)$$

$$T_2 = \mu(Epistemic) - 3\sigma(Epistemic) \quad (6)$$

For a given image if at least one uncertainty is greater than its corresponding threshold, we classify it as an adversarial image and pass it as input to our ensemble of iterative defenses. We chose the threshold values for T_1 and T_2 to be as shown in Eq. (5) and (6) because, although μ - σ is the least amount of uncertainty corresponding to annotated training dataset D, there could be unseen attacks in the real-world where the uncertainty value is below μ - σ. To accommodate these unseen adversarial images we set T_1 and T_2 to be two standard deviations lower than μ - σ as shown in Eq. (5) and (6).

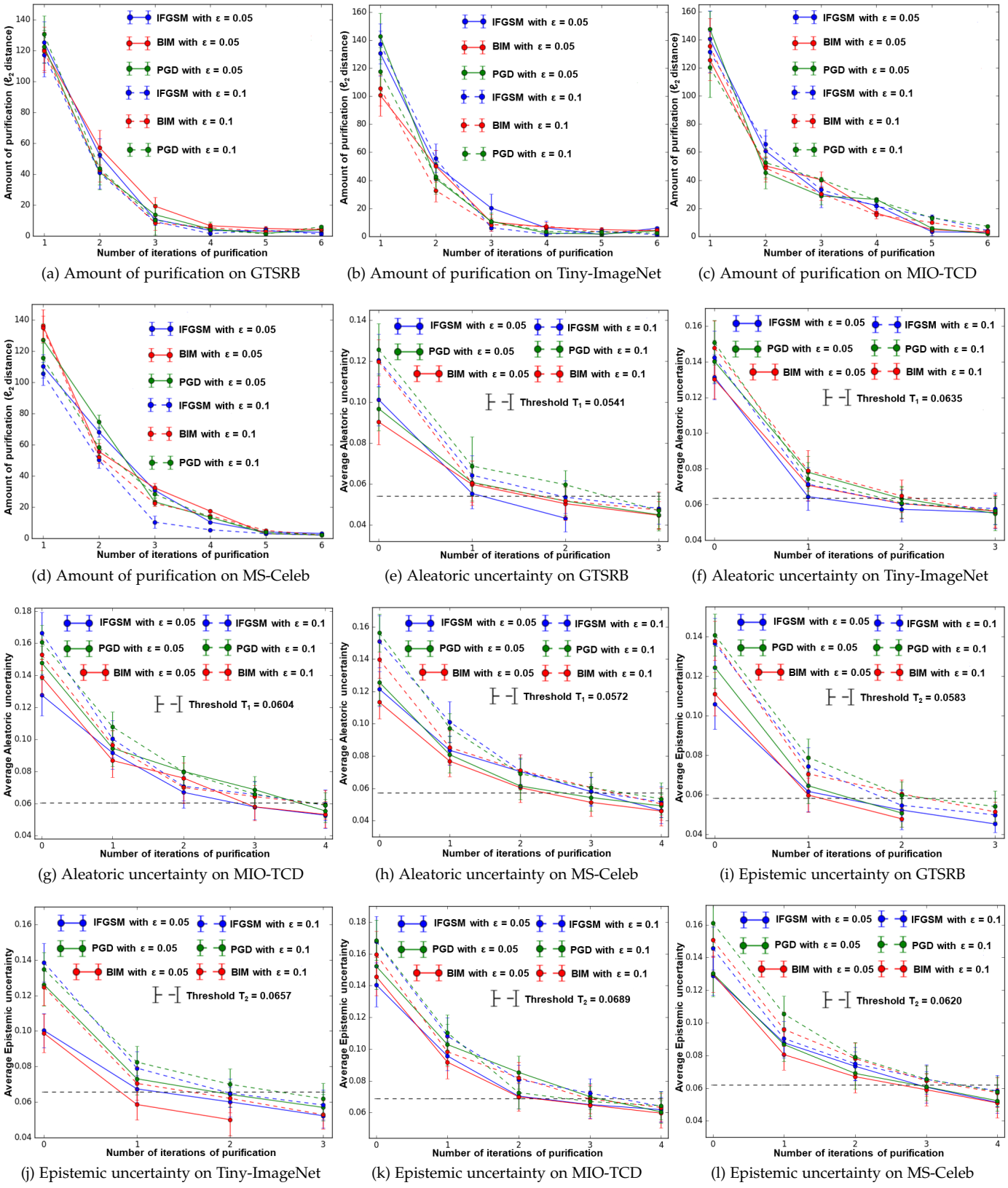
3.4 Ensemble of Iterative Adversarial Defenses

The adversarial defenses used in this paper are auxiliary generative networks that can be used in conjunction with any classifier as a pre-processing step without modifying the structure of the classifier. These approaches do not assume any classifier model and are model agnostic. In this paper we chose to use MagNet [26], PixelDefend [27], Shield-Nets [28], and Defense-GAN [101] in our ensemble because they achieve state-of-the-art results for white and black box defense. It should be noted that the above mentioned defenses are all single-step defenses and cannot quantify if a purified image is adversarial or not. However, using our defense framework, we are able to convert these single-step defenses into iterative defenses and quantify the amount of adversarial component remaining after each iteration of purification. Additionally, each of these individual defenses are trained independently and this provides flexibility to alter the structure of the ensemble, i.e., add/remove individual defenses without affecting the entire framework (see Section 4.5.1). Unlike the related work done for adversarial defense as shown in Table 1 and 2, we chose to use an ensemble of defenses for two reasons: 1) we show that an ensemble of defenses is able to statistically defend against adversarial attacks much better compared to individual defenses (See Section 4.5) and, 2) the output of the ensemble (the average purified image) acts as a momentum which helps prevent individual deterministic defenses from getting stuck at a local minimum [103].

3.4.1 Empirical Determination of the Number of Iterations for Purification

In Fig. 1, M is the maximum number of iterations an image can be purified before being (a) passed as input to the black box CNN or (b) rejected. The reason for this is that after each iteration, the amount of purification done decreases and after M iterations the ensemble is not able to further purify the image. This situation arises when the adversarial perturbation (ϵ) is very high causing the adversarial noise to dominate the image, which makes it very difficult to purify the image. This is a potential threat that an adversary could use to lock our defense in a state of an infinite loop of purification, thus crashing the defense framework. To eliminate this threat, we set a threshold (M) on the maximum number of iterations of purification before rejecting an image. In order to empirically determine the value of M, we attacked the *Substitute* CNN using the IFGSM [3], BIM [34], and PGD [100] attacks with ϵ = 0.05, 0.1, and 0.2. We chose the values of ϵ within the range of 0.05 - 0.2 because this is the range an adversarial attack is likely to fool a human observer and adversarial images with $\epsilon > 0.2$ make the resulting images more discernible to the human eye [28], [94]. The resulting adversarial images are then passed as input to our ensemble of image purifiers for six iterations of purification. From this we quantify the amount of purification done by measuring the L_2 distance between the input and output images at every iteration. Fig. 3 shows the plots for the amount of purification, Aleatoric, and Epistemic uncertainties Vs. the number of iterations of purification for the GTSRB [80], MIO-TCD [60], Tiny ImageNet [84], and MS-Celeb [59] datasets with ϵ = 0.05

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Amount of purification on GTSRB (b) Amount of purification on Tiny-ImageNet (c) Amount of purification on MIO-TCD

(d) Amount of purification on MS-Celeb (e) Aleatoric uncertainty on GTSRB (f) Aleatoric uncertainty on Tiny-ImageNet

(g) Aleatoric uncertainty on MIO-TCD (h) Aleatoric uncertainty on MS-Celeb (i) Epistemic uncertainty on GTSRB

(j) Epistemic uncertainty on Tiny-ImageNet (k) Epistemic uncertainty on MIO-TCD (l) Epistemic uncertainty on MS-Celeb

Fig. 3: (a) - (d) show the average amount of purification Vs. the number of iterations of purification, (e) - (h) show the average Aleatoric uncertainties and, (i) - (l) show the average Epistemic uncertainties for the GTSRB [80], MIO-TCD [60], Tiny-ImageNet [84], and MS-Celeb [59] datasets, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

and 0.1. From Fig. 3(a) - (d) we can see that after 3 iterations, the amount of purification does not significantly change for the GTSRB, and Tiny ImageNet datasets. Hence, we set the value of M = 3 for these datasets. Similarly, for the MIO-TCD and MS-Celeb datasets we set M = 4. It is also interesting to note that, as the dimension of the input space increases (see Table 5), the value of M also increases.

Note: Since the Bayesian uncertainties do not significantly change after M iterations (see Fig. 3), in order to save computational resources, we can purify all input images for M iterations rather than verifying after each iteration if the given image is still adversarial or not. By doing so, we also risk the possibility of purifying non-adversarial images and, this could affect the final classification (refer to Section 4.5.1). Additionally, without the adversarial detector we further invite the possibility of brute force attacks (i.e., attacks with severe adversarial perturbation). Prior work done in [94] shows that such images cannot be purified within M iterations and cause misclassifications, which in turn could potentially compromise safety critical automated applications. Ideally, such adversarial images would be rejected by the adversarial detector after M iterations of purification (see Fig. 2b in the supplementary material).

Hence after determining the value of M for a given dataset, applications that do not pose significant risk may not need the adversarial detector because the damage caused by brute force adversarial attacks and misclassifying non-adversarial images is negligible compared to safety critical automated applications such as remote surveillance [52] and human biometrics [115], where it is critical to have the adversarial detector.

3.4.2 Theoretical Lower Bound for the Amount of Purification in the Input space

For a given image $X \in \mathbb{R}^{P\times Q}$ from dataset D, where $P\times Q$ is the number of pixels in the image, an ϵ-bounded adversarial image is denoted as $X+\delta$, where δ belongs to the l_p bounded neighborhood $\Delta = \{\delta \in \mathbb{R}^{P\times Q} \mid \|\delta\|_p \leq \epsilon\}$ to X. The individual adversarial defenses $\pi_\omega(X'|X+\delta)$ are expected to map the adversarial images from adversarial regions back to a space within Δ, where ω are the trainable parameters of the defense and X' is the output of the defense. Adversarial attacks on any classification task with a loss function of $\mathscr{L}(X', Y; \theta)$, where, θ are the parameters of the black box classifier, can be achieved by optimizing,

$$\arg\max_{\delta\in\Delta} \int_\Delta \pi_\omega(X'|X+\delta)\, \mathscr{L}(X', Y; \theta)\mathrm{d}X'. \tag{7}$$

The loss function of the adversarial defense ($\mathscr{L}_{Def}$) can be expressed as the marginalized expectation:

$$\mathscr{L}_{Def} = \mathop{\mathbb{E}}_{(X,Y)\sim D} \int_\Delta \mathbb{E}_{X'\sim\pi_\omega(X'|X+\delta)}\big[\mathscr{L}(X', Y; \theta)\big] p(\delta)\mathrm{d}\delta \tag{8}$$

where $p(\delta)$ represents the distribution of adversarial samples in Δ. The theoretical possibility of the adversarial defense to neutralize the adversarial images is supported by the following theorem:

Theorem 1. *Assume $\mathscr{L}(X', Y; \theta)$ is continuous in $X+\Delta$ and $\pi_\omega(X'|X+\delta)$ is supported on $X+\Delta$, there exists a lower bound for $\mathscr{L}_{Def}$ in space Δ. If $\pi_\omega(X'|X+\delta) = \delta_{Dirac}(X'-X-\beta_0)$, $\mathscr{L}_{Def}$ reaches the lower bound, where $\beta_0 = \arg\min_{\beta\in\Delta} \mathscr{L}(X+\beta, Y; \theta)$.*

Proof of Theorem 1: Since $\mathscr{L}(X+\beta, Y; \theta)$ is continuous and Δ is compact, $\beta_0 = \arg\min_{\beta\in\Delta} \mathscr{L}(X+\beta, Y; \theta)$, where β_0 is the lower bound space within Δ beyond which an image cannot be further purified. Assume $p(\delta)$ is a distribution that only supports in Δ and π_ω supports in $X+\Delta$,

$$\mathscr{L}_{Def} = \mathop{\mathbb{E}}_{(X,Y)\sim D} \int_\Delta \mathbb{E}_{X'\sim\pi_\omega(X'|X+\delta)}\big[\mathscr{L}(X', Y; \theta)\big] p(\delta)\mathrm{d}\delta \tag{9}$$

$$= \mathop{\mathbb{E}}_{(X,Y)\sim D} \int_\Delta p(\delta)\mathrm{d}\delta \int_{X+\Delta} \mathrm{d}X'\pi_\omega(X' \mid X+\delta)\mathscr{L}(X', Y; \theta) \tag{10}$$

$$\geq \mathop{\mathbb{E}}_{(X,Y)\sim D} \int_\Delta p(\delta)\mathrm{d}\delta \int_{X+\Delta} \mathrm{d}X'\pi_\omega(X' \mid X+\delta) \tag{11}$$

$$= \mathop{\mathbb{E}}_{(X,Y)\sim D} \int_\Delta p(\delta)\mathrm{d}\delta \int_{X+\Delta} \mathrm{d}X'\pi_\omega(X' \mid X+\delta) \left(\min_{X'\in\Delta} \mathscr{L}(X', Y; \theta)\right) \mathscr{L}(X+\beta_0, Y; \theta) \tag{12}$$

$$= \mathop{\mathbb{E}}_{(X,Y)\sim D} \int_\Delta p(\delta)\mathrm{d}\delta \mathscr{L}(X+\beta_0, Y; \theta) \tag{13}$$

$$= \mathop{\mathbb{E}}_{(X,Y)\sim D} \mathscr{L}(X+\beta_0, Y; \theta) \tag{14}$$

The equality is satisfied when $\pi_\omega(X'|X+\delta) = \delta_{Dirac}(X'-X-\beta_0)$. Theorem 1 is further empirically supported by Fig. 3. In Fig. 3(a) - (d) it can be seen that after M iterations the amount of purification significantly decreases and the image can no longer be purified. This also means that after M iterations, the adversarial image X_{adv} is transformed/purified to $X' \in \beta_0$, beyond which it cannot be further purified. Additionally, from Fig. 3 it can be seen that as the amount of purification decreases after each iteration (Fig. 3(a) - (d)), the aleatoric and epistemic uncertainties also gradually decrease (Fig. 3(e) - (l)) and after M iterations they do not significantly vary because the amount of purification does not change. This further emphasizes that the resulting purified image X' has been reached the lower bound within the input space $\beta_0 \in \Delta$ and cannot be further purified.

4 Experimental Results

4.1 CNN Architectures

Table 4 shows the architectures of the CNN used in our approach for adversarial defense. For fair comparisons, the CNN architectures in Table 4 are the same as those reported in [28] and [94]. We evaluate our black box defense by creating an adversarial *Substitute* CNN and transfer the adversarial images generated for the adversary's *Substitute* as input to our defense [5], [99]. In Table 4 we used the same CNN architecture and training data for our defense's *Substitute* as well as the adversary's *Substitute*. By doing so

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 4: CNN architectures

CNN Architectures	
Target Black Box CNN	ResNet
Defense *Substitute* CNN	VGG
Bayesian CNN	Bayesian VGG
Adversary's *Substitute* CNN	VGG

we are giving the adversary an equal amount of knowledge as to our defense *Substitute* model in order to have a fair evaluation of our defense. Although we chose the architecture of the black box classifier to be ResNet [105], it should be noted that our framework in general can be adopted to other classifier as well. In fact, our prior work done in [94], shows results for a black box classifier with VGG architecture [106].

4.2 Datasets

We evaluate our defense on six public benchmark datasets namely: Fashion-MNIST [78], CIFAR-10 [79], GTSRB [80], MIO-TCD [60], Tiny ImageNet [84], and MS-Celeb [59]. Table 5 shows a summary of all the datasets and Fig. 4 shows examples of images from the six datasets used in this paper.

• ***Fashion-MNIST***: Fashion MNIST was designed to be a much more difficult and drop-in replacement for the MNIST dataset [87]. The dataset consists of 60,000 training and 10,000 testing gray-scale images of size 28x28 distributed evenly into 10 different classes.

• ***CIFAR-10***: CIFAR-10 is another widely used dataset that consists of 50,000 training and 10,000 testing RGB images of size 32x32 distributed evenly into 10 different classes.

• ***GTSRB***: The GTSRB dataset consists of 51,883 images of traffic sign in Germany. The dataset is split into 43 different classes with 39,295 training and 12,631 testing images.

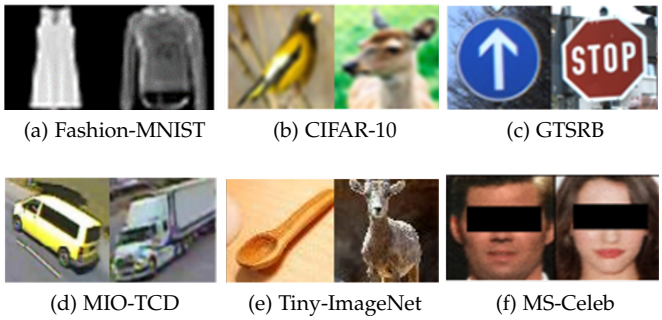
• ***MIO-TCD***: The MIO-TCD dataset consists of two sub-datasets namely: MIO-TCD Classification and MIO-TCD Localization. The dataset consists of images of vehicles captured from various traffic cameras installed on the roads all over the USA and Canada. The dataset is split into 11 classes with 519,164 training and 129,795 testing images for the classification subset. The localization subset is split into 11 classes with 110,000 training and 27,743 testing images.

• ***Tiny-ImageNet***: The Tiny-Imagenet consists of 200 classes with 500 training and 50 testing images per class. The 200 classes were collected from the Synsets of the WordNet [86] hierarchy and is similar to the ImageNet dataset [85].

• ***MS-Celeb***: The MS-Celeb dataset consists of approximately 9.5M images for 99,892 celebrities. It has been shown that this dataset is extremely noisy with many incorrect annotations [88] - [91]. In order to reduce the noise due to incorrect annotation, we followed the approach proposed by Jin *et al*. [93] which uses the community detection algorithm [92]. Based on this, the authors provided a list of correctly annotated images and showed that approximately 97.3% of images in the dataset are correctly labeled. This results in a total of approximately 6.5M images for 94,682 celebrities.

4.3 Threat Models

In this sub-section we define the adversarial attacks that are used for evaluating our defense. For a given test image-label pair (X, Y), adversarial attacks find a perturbation δ with $||\delta||_\infty \leq \epsilon$ such that a deep learning classifier $f(\cdot)$ results in $f(X+\delta) \neq Y$. ϵ is a hyper-parameter that sets the perturbation limit for each pixel in X on the color scale.



(a) Fashion-MNIST (b) CIFAR-10 (c) GTSRB

(d) MIO-TCD (e) Tiny-ImageNet (f) MS-Celeb

Fig. 4: Example of images from the (a) Fashion-MNIST [78], (b) CIFAR-10 [79], (c) GTSRB [80], (d) MIO-TCD [60], (e) Tiny-ImageNet [84], and (f) MS-Celeb [59] datasets, respectively. For ethical concerns and in order to preserve the identities of the celebrities in the MS-Celeb dataset we have masked the top portion of the faces.

• **Iterative Fast Gradient Sign Method (IFGSM)** [3]: This attack uses the sign of the gradients at every pixel to determine the direction of perturbation.

$$X_{n+1}^{adv} = X_n + \epsilon \cdot sign(\bigtriangledown_X L(X, Y)) \tag{15}$$

• **Basic Iterative Method (BIM)** [34]: This attack extends the FGSM attack [3] by iterating it multiple times with a small step size.

$$X_{n+1}^{adv} = Clip_\epsilon(X_n + \alpha \cdot sign(\bigtriangledown_X L(X_n, Y))) \tag{16}$$

• **Projected Gradient Descent** [100]: This attack computes the gradient in the direction of the highest loss and projects it back to the l_p norm around the sample.

$$X_{n+1}^{adv} = \prod^{\epsilon}(X_n + \alpha \cdot sign(\bigtriangledown_X L(X_n, y))) \tag{17}$$

In eq. (15) - (17), X^{adv} is the resulting adversarial image, $\bigtriangledown_X L(X, Y)$ is the loss function used to train the CNN, α is the iterative step size, Clip $(\cdot)$ and $\prod(\cdot)$ are the clipping and projection functions, respectively.

4.4 Performance Evaluation of the Proposed KD Approaches

In this sub-section we evaluate the KD-1 approach using the Fashion-MNIST [78], CIFAR-10 [79], and GTSRB [80] datasets. We evaluate KD-2 using the MIO-TCD [60] and Tiny-ImageNet [84] datasets and finally, we evaluate KD-3 using the MS-Celeb [59] dataset.

4.4.1 Knowledge Distillation on the MIO-TCD Dataset

This dataset consists of two parts: 1) classification dataset, and 2) localization dataset. We trained the *Teacher model* using the MIO-TCD classification dataset and used the localization dataset to create our pseudo-labeled dataset for training the *Student model*. It should be noted that in the MIO-TCD classification dataset, we ignored the class "Background" because this class does not belong in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 5: Summary of the datasets used in this paper.

Dataset	Image size	Domain	Grayscale/ RGB	Number of classes	Balanced classes?†	Training data	Testing data
Fashion-MNIST	28 x 28	Fashion	Grayscale	10	Yes (6,000)	60,000	10,000
CIFAR-10	32 x 32	General	RGB	10	Yes (5,000)	50,000	10,000
GTSRB	64 x 64*	German traffic signs	RGB	43	No	39,252	12,630
Tiny ImageNet	64 x 64	General	RGB	200	Yes (500)	100,000	10,000
MS-Celeb	128 x 128*	Celebrity faces	RGB	100	No	8,933	2,177
MIO-TCD	224 x 224**	USA & Canada vehicles	RGB	10	No	359,164	129,796

*images are resized to the specified size. ** shorter side of the image is resized to 256 while maintaining the aspect ratio, and then center is cropped to the size of 224 x 224. † The Number in parentheses indicates the number of images per class.

TABLE 6: Performance evaluation and comparison of our KD approaches with respect to the Teacher (black box) classifier.

Training dataset (Teacher classifier)	Testing dataset (Teacher & student classifier)	Teacher accuracy (%)	KD algorithm	Pseudo-labeled training dataset (student classifier)	Student accuracy (%)
Fashion-MNIST training dataset	Fashion-MNIST testing dataset	93.51	KD-1 (Z = 25%)	Z	77 .16 $\pm$ 1.55
			KD-1 (Z = 25%)	Z + GAN	89.67 $\pm$ 1.32
			KD-1 (Z = 50%)	Z + GAN	91.63 $\pm$ 0.76
CIFAR-10 training dataset	CIFAR-10 testing dataset	95.31	KD-1 (Z = 25%)	Z	58.16 $\pm$ 2.63
			KD-1 (Z = 25%)	Z + GAN	85.79 $\pm$ 1.78
			KD-1 (Z = 50%)	Z + GAN	88.42 $\pm$ 1.26
GTSRB training dataset	GTSRB testing dataset	96.45	KD-1 (Z = 25%)	Z	68.80 $\pm$ 3.07
			KD-1 (Z = 25%)	Z + GAN	90.34 $\pm$ 1.04
			KD-1 (Z = 50%)	Z + GAN	91.79 $\pm$ 0.68
MIO-TCD classification training dataset	MIO-TCD classification testing dataset	94.68	KD-2	MIO-TCD localization	84.51
			KD-2 + 50% overlap	MIO-TCD localization + 50% MIO-TCD classification	89.04 $\pm$ 0.83
Tiny ImageNet training dataset	Tiny ImageNet testing dataset	Top 1: 46.79 Top 5: 72.30	KD-2	200 classes from ImageNet training dataset	Top 1: 40.85 Top 5: 64.37
			KD-2 + 50% overlap	200 classes from ImageNet + 50% Tiny ImageNet	Top 1: 41.33 $\pm$ 0.57 Top 5: 65.59 $\pm$ 1.08
MS-Celeb 75% of 100 celebrities for training	25% of 100 celebrities for testing	90.32 $\pm$ 1.56	KD-3	Every other celebrity	74.58 $\pm$ 3.72
			KD-3 + 50% overlap	Every other celebrity + 50% of the 100 celebrities	76.40 $\pm$ 2.47

localization dataset [71]. Based on this we probed the *Teacher model* and created the pseudo-labeled dataset such that each class had at least 1,713 images.

4.4.2 Knowledge Distillation on the Tiny ImageNet Dataset

The Tiny ImageNet dataset consists of 200 classes and this dataset is used for training the *Teacher model* and the pseudo-labeled dataset is created using the ImageNet dataset [85]. It should be noted that we use only those corresponding 200 classes in the ImageNet dataset to create the pseudo-labeled dataset. Based on this we probed the *Teacher model* and created the pseudo-labeled dataset such that each class had at least 487 images

4.4.3 Knowledge Distillation on the MS-Celeb Dataset

In order to train the *Teacher model*, we manually selected 100 celebrities that had at least 100 images after discarding images that had extremely skewed poses and celebrities wearing sunglasses. A complete list of all of the selected facial identities used for training the *Teacher model* is provided in the supplementary material. We denote this dataset as $Q_{1:100}$ (Q_i is the identity of the celebrity) and it is used for training the *Teacher model*. In order to train the *Student model* we first create a pseudo-labeled dataset by probing the *Teacher model* with images of celebrities that do not belong in $Q_{1:100}$ and labeled the images with the predicted class. We denote this pseudo-labeled dataset as $Q_{101:\infty}$ and it should be noted that the dataset $Q_{1:100}$ and $Q_{101:\infty}$ contain images of different celebrities and their data distributions do not overlap (ignoring the noise due to incorrect annotations). Based on this we augmented 3,000 pseudo-labeled images per class in the $Q_{101:\infty}$ dataset. In the MIO-TCD and MS-Celeb datasets we chose the number of augmented images in the pseudo-labeled dataset to be at least 1,713 and 487 images, per class, respectively, because this is the maximum number of images possible for the class with the least number of images. Table 6 shows the baseline performance comparison between the *Student model* (defense substitute classifier) and the *Teacher model* (black box classifier). In Table 6 although the *Teacher model* outperforms the *Student model*, it can be seen that as the overlap between the *Teacher model's* training dataset and the *student model's* pseudo-labeled dataset increases, the performance of the *Student model* also increases. For the following experimental results in Section 4.5, we train our defense framework to defend the *Student model* in Table 6 and then deploy the trained defense framework to defend the black box classifier.

4.5 Performance Evaluation of Adversarial Detection using Bayesian Uncertainties

Table 7 shows the performance evaluation of our adversarial detection using Bayesian uncertainties. For the results in Table 7 we created adversarial images using the IFGSM attack [3] with ϵ = 0.05 and 0.2 and passed both the adversarial and non-adversarial images as inputs to the Bayesian CNN. From Table 7 when ϵ = 0.05, we can see that our approach achieves at least 69.05% accuracy on the Tiny-ImageNet dataset [84] and at most 93.51% on the Fashion-MNIST dataset [78]. The reason for this is that the Fashion-MNIST dataset contains images with one foreground object in a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 7: Performance evaluation of adversarial detection using Bayesian uncertainties.

Dataset	Accuracy (%)	False Positive (%)	False Negative (%)
IFGSM attack with ϵ = 0.05			
F-MNIST	93.51	3.95	2.54
CIFAR-10	79.34	15.26	5.40
GTSRB	91.06	6.36	2.58
MIO-TCD	87.74	9.77	2.49
Tiny- ImageNet	69.05	23.94	7.01
MS-Celeb	72.93	20.15	6.92
IFGSM attack with ϵ = 0.2			
F-MNIST	98.34	1.13	0.53
CIFAR-10	95.47	3.65	0.88
GTSRB	97.61	1.38	1.01
MIO-TCD	97.03	2.16	0.81
Tiny-ImageNet	92.58	5.35	2.07
MS-Celeb	93.45	4.91	1.64

completely black background compared to Tiny-ImageNet which has a more complex background (please refer the supplementary material for visual comparisons on all of the datasets). Furthermore, in Table 7 when ϵ = 0.05, the false positive is significantly higher than the false negative, meaning that more non-adversarial images were predicted as adversarial images than vice-versa. In the context of adversarial detection it is better to have a higher false positive rate compared to false negative rate because the damage caused by misclassifying an adversarial image is more than misclassifying a non-adversarial image [112] - [114]. Moreover, as the adversarial perturbation (ϵ) increases from 0.05 to 0.2, the Signal-to-Noise ratio (SNR) of the image starts to decrease making the adversarial noise more easily visible to the human eye as well as easily detected using the Bayesian CNN [94]. As shown in Fig. 3(a) - 3(i), after M iterations of purification, these images would automatically be rejected by the system if the predicted uncertainty values are above the corresponding thresholds.

4.5.1 How does purifying non-adversarial images affect the classification?

In this sub-section we perform experiments to observe how purifying the non-adversarial images affect the final classification. Table 8 shows the total number of non-adversarial images (i.e., images that failed to fool the black box classifier) using the IFGSM attack with ϵ = 0.05 and the total number of images predicted correctly after one iteration of purification.

TABLE 8: Performance evaluation of purifying non-adversarial images.

Dataset	Total No. of non-adversarial images	Images correctly classified after purification*
F-MNIST	3920	3871 (98.8%)
CIFAR-10	2617	2588 (98.9%)
GTSRB	4042	3965 (98.1%)
Tiny-ImageNet	1276	1257 (98.5%)
MIO-TCD	26439	26305 (99.5%)
MS-Celeb	562	547 (97.4%)

*corresponds to images correctly classified with respect to the predictions of the black box before and after purification

In Table 8, a correctly classified image is defined as: for a given image X, the predicted label of black box f is Y, (i.e. $f(X) = y$). The corresponding purified image X' has a black box predicted label Y', (i.e. $f(X') = Y'$). If $Y = Y'$ then the image is correctly classified, but it should be noted that Y and Y' do not necessarily correspond to the human annotated ground-truth. From Table 8 we can see that the least possible accuracy is 97.4% which indicates that even after one iteration of purification, the non-adversarial images are not significantly altered to cause severe misclassification.

4.6 Performance Evaluation and Comparison of our Defense Against Adversarial Attacks

Tables 9 - 14 show the performance and comparison of our approach against the state-of-the-art on the GTSRB [80], Tiny-ImageNet [84], MIO-TCD [60], and MS-Celeb [59] datasets described in Section 4.2. Tables 1 - 4 in the supplementary material show the performance and comparison of our approach on the Fashion-MNIST [78] and CIFAR-10 [79] datasets. Note that in Tables 9 - 14 all of the single-step adversarial defenses (MagNet [26], PixelDefend [27], ShieldNets [56], and Defense-GAN [101]) have been converted into iterative defenses using our framework. From Tables 9 - 14 it can be seen that our ensemble of defenses outperforms all the stand-alone defenses. Although, as the perturbation limit (ϵ) of the adversarial attack increases, the performance of all the approaches in Tables 9 - 14 gradually decrease, but there is also a gradual increase in the Bayesian uncertainty metrics. This means that even if an adversary tries to break our defense by significantly increasing the value of (ϵ), the resulting adversarial image would still be rejected because the uncertainty values of the image are beyond the threshold limits T_1 and T_2. By increasing the value of (ϵ) in order to break a defense may seem trivial, however, it is still a foundational problem in online applications where there is no human in the loop [38], [39]. In Table 10, we report the Top 5 classification accuracy for the Tiny ImageNet as this is the metric that is used for evaluating the dataset. Additionally, the adversarial images used in Table 10 were generated such that the top 5 predictions for an adversarial image do not contain the ground-truth label (Y), i.e., $Y \notin \{Y'_{i=1:5}\}$.

4.6.1 Ablation Study for Evaluating Different Combinations of Ensembles of Iterative Defenses

In this sub-section we perform an ablation study to evaluate different combinations of ensembles of adversarial defenses and compare their performance. For this purpose we chose to use the Fashion-MNIST [78] and CIFAR-10 [79] datsets and the following adversarial defenses: MagNet [26], PixelDefend [27], ShieldNets [56], and Defense-GAN [101]. Table 15 shows the comparisons of different ensembles.

From Table 15 it can be seen that using the ensemble MPSD and MSD achieves the best classification accuracy on the Fashion-MNIST dataset against the IFGSM and BIM attack, respectively. Similarly, using the ensemble MPS achieves the best classification accuracy on the CIFAR-10 dataset against the IFGSM and BIM attacks. It is also interesting to note that although some of the combinations of ensembles such as MS on the CIFAR-10 dataset and MSD on the Fashion-MNIST dataset had lower aleatoric and epistemic uncertainties compared to MPS and MPSD, respectively, their classification accuracy was lower by at

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 9: Performance comparison of our Defense on the GTSRB dataset using the KD-1 approach with Z = 25% and 50%.

		ϵ = 0.1 (26/255)					
		GTSRB KD 1 (Z = 25%)			GTSRB KD 1 (Z = 50%)		
			T_1 = 0.0541	T_2 = 0.0583		T_1 = 0.0525	T_2 = 0.0570
Attack	**Defense**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**
IFGSM	**No Defense**	28.93	-	-	28.93	-	-
	MagNet	83.50 ± 3.47	0.0527 ± 0.0093	0.0570 ± 0.0084	84.01 ± 3.12	0.0497 ± 0.0091	0.0515 ± 0.0087
	ShieldNets	86.19 ± 3.20	0.0493 ± 0.0079	0.0513 ± 0.0074	86.89 ± 3.35	0.0477 ± 0.0080	0.0483 ± 0.0077
	PixelDefend	76.27 ± 2.68	0.0561 ± 0.0081	0.0584 ± 0.0085	79.43 ± 2.81	0.0537 ± 0.0084	0.0534 ± 0.0088
	Ensemble	**89.26 ± 3.07**	**0.0481 ± 0.0076**	**0.0462 ± 0.0083**	**89.94 ± 2.74**	**0.0467 ± 0.0090**	**0.0471 ± 0.0081**
BIM	**No Defense**	22.05	-	-	22.05	-	-
	MagNet	81.79 ± 3.28	0.0533 ± 0.0080	0.0564 ± 0.0084	81.33 ± 3.76	0.0539 ± 0.0074	0.0548 ± 0.0079
	ShieldNets	84.25 ± 3.79	0.0487 ± 0.0087	0.0526 ± 0.0082	86.07 ± 3.16	0.0459 ± 0.0081	0.0502 ± 0.0078
	PixelDefend	79.03 ± 3.14	0.0544 ± 0.0077	0.0558 ± 0.0093	80.76 ± 2.91	0.0532 ± 0.0085	0.0537 ± 0.0081
	Ensemble	**86.18 ± 3.44**	**0.0473 ± 0.0089**	**0.0485 ± 0.0081**	**86.97 ± 3.03**	**0.0482 ± 0.0083**	**0.0460 ± 0.0095**
PGD	**No Defense**	24.51	-	-	24.51	-	-
	MagNet	82.83 ± 3.81	0.0511 ± 0.0086	0.0541 ± 0.0074	83.44 ± 3.17	0.0518 ± 0.0070	0.0510 ± 0.0073
	ShieldNets	86.42 ± 3.28	0.0465 ± 0.0078	0.0513 ± 0.0090	86.95 ± 3.47	0.0472 ± 0.0093	0.0496 ± 0.0078
	PixelDefend	80.43 ± 3.50	0.0521 ± 0.0080	0.0531 ± 0.0092	82.04 ± 2.93	0.0503 ± 0.0088	0.0508 ± 0.0080
	Ensemble	**88.41 ± 3.28**	**0.0496 ± 0.0083**	**0.0470 ± 0.0088**	**89.40 ± 3.53**	**0.0451 ± 0.0090**	**0.0457 ± 0.0074**

TABLE 10: Performance comparison of our Defense on the Tiny ImageNet dataset using the KD-2 approach.

		ϵ = 0.1 (26/255)					
		Black box training data = Tiny ImageNet Defense training data = ImageNet			Black box training data = Tiny ImageNet Def. training data = ImageNet + 50% Tiny ImageNet		
			T_1 = 0.0635	T_2 = 0.0657		T_1 = 0.0598	T_2 = 0.0647
Attack	**Defense**	**Top 5 Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**	**Top 5 Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**
IFGSM	**No Defense**	12.76	-	-	12.76	-	-
	MagNet	42.66	0.0587 ± 0.0097	0.0645 ± 0.0107	44.57 ± 1.24	0.0523 ± 0.0089	0.0607 ± 0.0095
	ShieldNets	45.83	0.0543 ± 0.0086	0.0600 ± 0.0095	47.17 ± 1.06	0.0527 ± 0.0082	0.0571 ± 0.0088
	PixelDefend	38.29	0.0622 ± 0.0103	0.0651 ± 0.0089	40.07 ± 0.97	0.0591 ± 0.0099	0.0605 ± 0.0106
	Ensemble	**46.34**	**0.0577 ± 0.0088**	**0.0562 ± 0.0083**	**50.34 ± 1.08**	**0.0528 ± 0.0080**	**0.0526 ± 0.0078**
BIM	**No Defense**	14.97	-	-	14.97	-	-
	MagNet	45.29	0.0535 ± 0.0081	0.0581 ± 0.0079	45.80 ± 0.87	0.0521 ± 0.0090	0.0569 ± 0.083
	ShieldNets	47.93	0.0522 ± 0.0085	0.0546 ± 0.0093	49.27 ± 1.11	0.0508 ± 0.0079	0.0518 ± 0.0086
	PixelDefend	41.75	0.0570 ± 0.0097	0.0636 ± 0.0102	42.39 ± 0.72	0.0560 ± 0.0108	0.0627 ± 0.0094
	Ensemble	**49.02**	**0.0560 ± 0.0092**	**0.0529 ± 0.0079**	**50.86 ± 1.04**	**0.0483 ± 0.0091**	**0.0502 ± 0.0077**
PGD	**No Defense**	8.24	-	-	8.24	-	-
	MagNet	40.17	0.0609 ± 0.0108	0.0638 ± 0.0091	41.85 ± 1.37	0.0587 ± 0.0110	0.0611 ± 0.0087
	ShieldNets	41.28	0.0574 ± 0.0090	0.0625 ± 0.0098	43.18 ± 1.30	0.0571 ± 0.0103	0.0604 ± 0.0093
	PixelDefend	35.16	0.0641 ± 0.0105	0.0683 ± 0.0092	37.90 ± 1.45	0.0635 ± 0.0097	0.0667 ± 0.0094
	Ensemble	**42.86**	**0.0566 ± 0.0082**	**0.0619 ± 0.0086**	**44.01 ± 1.08**	**0.0534 ± 0.0080**	**0.0579 ± 0.0089**

TABLE 11: Performance comparison of our Defense on the MIO-TCD classification dataset using the KD-2 approach with no overlap between the black box and pseudo-labeled training dataset.

		Black box training data = MIO-TCD Classification; Defense training data = MIO-TCD Localization					
		ϵ = 0.05 (13/255)			ϵ = 0.1 (26/255)		
			T_1 = 0.0604	T_2 = 0.0689		T_1 = 0.0604	T_2 = 0.0689
Attack	**Defense**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**
IFGSM	**No Defense**	20.37	-	-	13.52	-	-
	MagNet	83.65	0.0548 ± 0.0097	0.0609 ± 0.0094	73.28	0.0588 ± 0.0094	0.0637 ± 0.0102
	ShieldNets	84.51	0.0533 ± 0.0104	0.0588 ± 0.0087	75.89	0.0579 ± 0.0089	0.0618 ± 0.0096
	PixelDefend	75.06	0.0583 ± 0.0092	0.0645 ± 0.0095	69.57	0.0632 ± 0.0085	0.0683 ± 0.0094
	Ensemble	**85.80**	**0.0527 ± 0.0082**	**0.0568 ± 0.0096**	**77.04**	**0.0547 ± 0.0082**	**0.0607 ± 0.0079**
BIM	**No Defense**	18.36	-	-	14.21	-	-
	MagNet	82.79	0.0539 ± 0.0110	0.0575 ± 0.0093	74.01	0.0576 ± 0.0091	0.0621 ± 0.0084
	ShieldNets	81.08	0.0557 ± 0.0089	0.0571 ± 0.0076	72.60	0.0589 ±0.0083	0.0635 ± 0.0105
	PixelDefend	76.92	0.0569 ± 0.0087	0.0622 ± 0.0108	67.83	0.0645 ± 0.0095	0.0702 ± 0.0090
	Ensemble	**84.16**	**0.0512 ± 0.0084**	**0.0545 ± 0.0091**	**74.58**	**0.0559 ± 0.0084**	**0.0613 ± 0.097**
PGD	**No Defense**	15.89	-	-	12.76	-	-
	MagNet	80.55	0.0570 ± 0.0095	0.0598 ± 0.0084	73.96	0.0569 ± 0.0104	0.0602 ± 0.0094
	ShieldNets	81.25	0.0549 ± 0.0078	0.0601 ± 0.0103	75.07	0.0572 ± 0.0087	0.0626 ± 0.0097
	PixelDefend	73.39	0.0597 ± 0.0089	0.0640 ± 0.0097	65.72	0.0652 ± 0.0086	0.0733 ± 0.0081
	Ensemble	**82.48**	**0.0537 ± 0.0088**	**0.0606 ± 0.0084**	**75.78**	**0.0560 ± 0.0091**	**0.0623 ± 0.0084**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 12: Performance comparison of our Defense on the MIO-TCD classification dataset using the KD-2 approach with 50% overlap between the black box and pseudo-labeled training dataset.

		Black box training data = MIO-TCD Classification Defense training data = MIO-TCD Localization + 50% of the MIO-TCD classification					
		ϵ = 0.05 (13/255)			**ϵ = 0.1 (26/255)**		
			T_1 = 0.0580	**T_2 = 0.0656**		**T_1 = 0.0580**	**T_2 = 0.0656**
Attack	**Defense**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**
IFGSM	**No Defense**	20.37	-	-	13.52	-	-
	MagNet	83.25 ± 2.18	0.0520 ± 0.0081	0.0569 ± 0.0090	74.56 ± 1.35	0.0563 ± 0.0080	0.0580 ± 0.0084
	ShieldNets	85.02 ± 1.43	0.0521 ± 0.0082	0.0562 ± 0.0082	77.24 ± 1.46	0.0546 ± 0.0081	0.0579 ± 0.0078
	PixelDefend	76.81 ± 2.06	0.0554 ± 0.0076	0.0592 ± 0.0082	70.59 ± 1.22	0.0617 ± 0.0087	0.0633 ± 0.0091
	Ensemble	**86.39 ± 1.56**	**0.0513 ± 0.0072**	**0.0520 ± 0.0081**	**78.59 ± 1.50**	**0.0508 ± 0.0090**	**0.0577 ± 0.0073**
BIM	**No Defense**	18.36	-	-	14.21	-	-
	MagNet	84.58 ± 1.87	0.0501 ± 0.0082	0.0543 ± 0.0085	75.83 ± 1.93	0.0548 ± 0.0074	0.0591 ± 0.0082
	ShieldNets	84.27 ± 2.05	0.0519 ± 0.0083	0.0536 ± 0.0079	74.09 ± 1.36	0.0565 ± 0.0086	0.0604 ± 0.0088
	PixelDefend	77.17 ± 1.91	0.0548 ± 0.0080	0.0583 ± 0.0081	69.53 ± 1.84	0.0618 ± 0.0092	0.0648 ± 0.0083
	Ensemble	**86.58 ± 1.54**	**0.0494 ± 0.0078**	**0.0511 ± 0.0082**	**76.17 ± 1.20**	**0.0561 ± 0.0076**	**0.0585 ± 0.086**
PGD	**No Defense**	15.89	-	-	12.76	-	-
	MagNet	81.76 ± 1.51	0.0533 ± 0.0091	0.0556 ± 0.0077	75.15 ± 1.08	0.0552 ± 0.0084	0.0608 ± 0.0079
	ShieldNets	82.67 ± 1.58	0.0530 ± 0.0072	0.0539 ± 0.0082	75.68 ± 1.27	0.0573 ± 0.0096	0.0592 ± 0.0077
	PixelDefend	75.02 ± 1.76	0.0560 ± 0.0080	0.0574 ± 0.0082	68.18 ± 1.33	0.0606 ± 0.0073	0.0631 ± 0.0088
	Ensemble	**83.92 ± 1.51**	**0.0527 ± 0.0084**	**0.0563 ± 0.0087**	**78.13 ± 1.45**	**0.0515 ± 0.0082**	**0.0541 ± 0.0092**

TABLE 13: Performance comparison of our Defense on the MS-Celeb dataset using the KD-3 approach with no overlap between the black box and pseudo-labeled training dataset.

		Black box training dataset = 75% of the 100 celebrities Defense training dataset = All celebrities other than the 100 celebrities					
		ϵ = 0.05 (13/255)			**ϵ = 0.1 (26/255)**		
			T_1 = 0.0572	**T_2 = 0.0620**		**T_1 = 0.0572**	**T_2 = 0.0620**
Attack	**Defense**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**
IFGSM	**No Defense**	25.81 ± 3.78	-	-	18.77 ± 4.29	-	-
	MagNet	65.79 ± 3.57	0.0493 ± 0.0093	0.0547 ± 0.0087	61.38 ± 2.91	0.0541 ± 0.0068	0.0585 ± 0.0077
	ShieldNets	66.93 ± 3.02	0.0475 ± 0.0081	0.0503 ± 0.0098	62.02 ± 2.75	0.0539 ± 0.0079	0.0546 ± 0.0087
	PixelDefend	60.03 ± 3.29	0.0536 ± 0.0068	0.0579 ± 0.0074	58.15 ± 3.04	0.0551 ± 0.0080	0.0613 ± 0.0092
	Ensemble	**70.38 ± 2.64**	**0.0416 ± 0.0072**	**0.0472 ± 0.0064**	**63. 44 ± 2.70**	**0.0506 ± 0.0065**	**0.0542 ± 0.0071**
BIM	**No Defense**	29.55 ± 4.16	-	-	22.43 ± 3.97	-	-
	MagNet	63.84 ± 3.34	0.0507 ± 0.0075	0.0536 ± 0.0083	61.97 ± 3.24	0.0522 ± 0.0089	0.0576 ± 0.0082
	ShieldNets	66.45 ± 3.59	0.0468 ± 0.0077	0.0529 ± 0.0092	60.55 ± 2.74	0.0567 ± 0.0068	0.0566 ± 0.0072
	PixelDefend	62.76 ± 3.40	0.0495 ± 0.0071	0.0530 ± 0.0088	56.38 ± 3.39	0.0583 ± 0.0083	0.0637 ± 0.0090
	Ensemble	**68.91 ± 3.17**	**0.0459 ± 0.0082**	**0.0498 ± 0.0064**	**62.84 ± 2.13**	**0.0517 ± 0.0069**	**0.0528 ± 0.0074**
PGD	**No Defense**	22.03 ± 4.55	-	-	17.94 ± 4.25	-	-
	MagNet	61.93 ± 3.48	0.0526 ± 0.0091	0.0562 ± 0.0085	60.37 ± 2.61	0.0553 ± 0.0079	0.0557 ± 0.0082
	ShieldNets	64.05 ± 2.77	0.0489 ± 0.0083	0.0510 ± 0.0094	62.58 ± 2.84	0.0534 ± 0.0082	0.0540 ± 0.0073
	PixelDefend	59.10 ± 2.11	0.0544 ± 0.0089	0.0570 ± 0.0090	55.93 ± 3.07	0.0606 ± 0.0097	0.0618 ± 0.0103
	Ensemble	**65.78 ± 2.91**	**0.0491 ± 0.0073**	**0.0523 ± 0.0078**	**62.97 ± 2.71**	**0.0535 ± 0.0068**	**0.0529 ± 0.0076**

TABLE 14: Performance comparison of our Defense on the MS-Celeb dataset using the KD-3 approach with 50% overlap between the black box and pseudo-labeled training dataset.

		Black box training dataset = 75% of the 100 celebrities Def. training dataset = All other celebrities other than the 100 celebrities + 50% of black box training dataset					
		ϵ = 0.05 (13/255)			**ϵ = 0.1 (26/255)**		
			T_1 = 0.0558	**T_2 = 0.0594**		**T_1 = 0.0558**	**T_2 = 0.0594**
Attack	**Defense**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**
IFGSM	**No Defense**	25.81 ± 3.78	-	-	18.77 ± 4.29	-	-
	MagNet	67.03 ± 3.04	0.0444 ± 0.0084	0.0519 ± 0.0080	62.97 ± 2.26	0.0448 ± 0.0084	0.0549 ± 0.0094
	ShieldNets	67.58± 2.80	0.0438 ± 0.0084	0.0466 ± 0.0091	65.41 ± 2.48	0.0425 ± 0.0070	0.0490 ± 0.0091
	PixelDefend	63.57 ± 2.49	0.0485 ± 0.0072	0.0536 ± 0.0078	59.71 ± 3.17	0.0515 ± 0.0089	0.0533 ± 0.0096
	Ensemble	**71.82 ± 2.94**	**0.0387 ± 0.0071**	**0.0479 ± 0.0077**	**65. 73 ± 2.62**	**0.0437 ± 0.0081**	**0.0476 ± 0.0075**
BIM	**No Defense**	29.55 ± 4.16	-	-	22.43 ± 3.97	-	-
	MagNet	68.70 ± 2.92	0.0424 ± 0.0070	0.0507 ± 0.0085	62.74 ± 3.19	0.0497 ± 0.0085	0.0523 ± 0.0079
	ShieldNets	69.71 ± 2.83	0.0402 ± 0.0082	0.0445 ± 0.0071	64.04 ± 2.80	0.0478 ± 0.0098	0.0541 ± 0.0092
	PixelDefend	66.04 ± 3.12	0.0452 ± 0.0074	0.0467 ± 0.0083	61.95 ± 2.38	0.0538 ± 0.0097	0.0540 ± 0.0103
	Ensemble	**73.96 ± 2.18**	**0.0371 ± 0.0088**	**0.0431± 0.0072**	**64.37 ± 2.59**	**0.0458 ± 0.0086**	**0.0514 ± 0.0079**
PGD	**No Defense**	22.03 ± 4.55	-	-	17.94 ± 4.25	-	-
	MagNet	65.10 ± 2.75	0.0460 ± 0.0087	0.0511 ± 0.0079	60.06 ± 3.35	0.0544 ± 0.0093	0.0564 ± 0.0108
	ShieldNets	68.37 ± 2.36	0.0434 ± 0.0081	0.0478 ± 0.0072	61.64 ± 2.53	0.0507 ± 0.0095	0.0548 ± 0.0091
	PixelDefend	62.89 ± 3.14	0.0483 ± 0.0079	0.0547 ± 0.0093	57.80 ± 1.94	0.0572 ± 0.0090	0.0586 ± 0.0097
	Ensemble	**70.61 ± 2.70**	**0.0411 ± 0.0079**	**0.0496 ± 0.0070**	**62.50 ± 2.40**	**0.0469 ± 0.0083**	**0.0532 ± 0.0086**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3125931, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 15: Ablation study for comparing different combinations of ensembles of iterative defenses.

		IFGSM ϵ = 0.05 (13/255)			BIM ϵ = 0.05 (13/255)		
			T_1 = 0.0437	T_2 = 0.0624		T_1 = 0.0437	T_2 = 0.0624
Dataset	**Ensemble***	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**	**Accuracy (%)**	**Avg. Aleatoric uncertainty**	**Avg. Epistemic uncertainty**
Fashion-MNIST	**MPD**	84.33 ± 4.01	0.0314 ± 0.0079	0.0347 ± 0.0080	88.07 ± 3.72	0.0291 ± 0.0084	0.0340 ± 0.0076
	MSD	86.83 ± 4.44	0.0276 ± 0.0085	0.0336 ± 0.0086	**89.68 ± 3.44**	0.0253 ± 0.0082	**0.0307 ± 0.0079**
	PSD	85.09 ± 5.07	0.0304 ± 0.0077	0.0352 ± 0.0078	87.45 ± 4.03	0.0286 ± 0.0086	0.0348 ± 0.0089
	MPS	87.35 ± 3.24	0.0309 ± 0.0083	0.0335 ± 0.0081	88.19 ± 3.95	0.0267 ± 0.0070	0.0328 ± 0.0074
	MPSD	**87.94 ± 4.51**	**0.0288 ± 0.0089**	**0.0324 ± 0.0091**	89.32 ± 3.35	**0.0247 ± 0.0079**	0.0319 ± 0.0089
			T_1 = 0.0597	T_2 = 0.0683		T_1 = 0.0597	T_2 = 0.0683
CIFAR-10	**MP**	75.08 ± 4.49	0.0513 ± 0.0078	0.0546 ± 0.0083	75.66 ± 3.57	0.0515 ± 0.0079	0.0503 ± 0.0084
	MS	77.23 ± 4.05	**0.0472 ± 0.0086**	0.0539 ± 0.092	77.91 ± 3.11	0.0530 ± 0.0081	**0.0469 ± 0.0094**
	PS	74.37 ± 3.64	0.0533 ± 0.0077	0.0560 ± 0.0088	78.27 ± 3.80	0.0529 ± 0.0093	0.0494 ± 0.0089
	MPS	**77.86 ± 3.87**	0.0487 ± 0.0095	**0.0521 ± 0.0081**	**78.91 ± 3.28**	**0.0501 ± 0.0087**	0.0517 ± 0.0090

*M, P, S, and D refers to MagNet, PixelDefend, ShieldNets, and Defense-GAN

TABLE 16: Performance comparison of our defense against six black box adversarial attacks.

Dataset	**SimBA** [102]	**NES** [107]	$\mathcal{N}$**attack** [108]	**SPSA** [109]	**Boundary** [110]	**ZOO** [111]
GTSRB	22.73/**83.07**	27.02/**80.18**	16.84/**74.60**	23.79/**85.04**	17.47/**78.92**	25.40/**80.39**
MS-Celeb	17.06/**62.91**	16.54/**60.29**	11.35/**55.72**	19.18/**59.67**	15.11/**58.09**	18.58/**63.90**
CIFAR-10	20.47/**64.80**	24.58/**66.19**	15.57/**58.25**	21.27/**61.06**	18.71/**56.73**	23.89/**64.33**

Each cell is represented as x/y where x and y are the accuracy before and after applying our ensemble of defenses (i.e., MagNet + ShieldNets + PixelDefend).

least 1%. This observation indicates that although some purified images have slightly above average uncertainty metrics does not necessarily mean that these images are adversarial. A possible reason for this phenomenon is that since the Bayesian CNN is trained independently with almost no knowledge of the black box classifier's weights or its training dataset, the features learned by these two CNNs could be slightly different from each other. Hence, an image that is adversarial to the Bayesian CNN may not necessarily be adversarial to the black box classifier. It should also be noted that this kind of situation is statistically more likely to happen only to the images that are close to the classification boundary in the CNN's feature space and it is a very small percentage of the total dataset.

4.6.2 Performance against Black box Adversarial Attacks

In this sub-section we evaluate our defense against six black box adversarial attacks SimBA [102], NES [107], $\mathcal{N}$attack [108], SPSA [109], Boundary [110], and ZOO [111] . NES [107] and SPSA [109] attacks use query limited attacks where the attacker has a query budget L, within which the adversarial attack needs to be successful. The authors replace the gradient of the loss function with an estimate of the gradient, which is approximated by querying the classifier rather than computed by auto-differentiation. SimBA [102] is similar to NES attack [107] but rather than traversing in a specific direction of the gradient, they pick a random direction from a pre-specified set of orthogonal search directions, use the probability logit scores to check if it is pointing towards or away from the decision boundary, and perturb the image by adding the vector from the image. ZOO [111] estimates the gradient at each coordinate by finite differences and adopts C&W [4] for attacks based on the estimated gradient. $\mathcal{N}$attack [108] does not estimate the gradient but learns a Gaussian distribution centered around the input such that a sample drawn from it is likely adversarial. A major difference between our defense and SimBA [102] is that our approach does not need any probability logit output and requires only the final class prediction, but for the sake of comparison we provide the logit probabilities as input.

Table 16 shows the performance of our defense against the six black box attacks on the GTSRB [80], MS-Celeb [59], and CIFAR-10 [79] datasets. In Table 16 we set the query limit L for all query based attacks to 1,000 queries per image and adversarial perturbation limit ϵ = 0.1. We created the black box adversarial attacks by initially attacking our black box classifier without any defense. After creating the adversarial attacks we then transfer these images to the ensemble of adversarial defenses for a maximum of M iterations as shown in Fig. 1 and compute the accuracy. From Table 16 we can see that without our defense the lowest possible accuracy is 16.84%, 11.35%, and 15.57% on the GTSRB, MS-Celeb and CIFAR-10 datasets, respectively. With our defense we can significantly improve the performance to at least 74.60%, 55.72%, and 56.73% accuracy on the GTSRB, MS-Celeb and CIFAR-10 datasets, respectively which is an improvement of at least 3.5x.

4.7 Robustness of Defense - an Adversary's Point of View

In Section 3.1, we assumed that the adversary has no knowledge about our defense and can only see the input and final classification output of the black box classifier. Although this may seem to be a strong assumption, in this subsection we relax this assumption by allowing the adversary to have partial amounts of information about our defense. We evaluated the CIFAR-10 dataset against the IFGSM and PGD attack with ϵ = 0.1 in order to compare our defense with MagNet [26], PixelDefend [27] and ShieldNets [28]. We quantify the robustness of our approach by computing the time taken for the adversary to create 50 successful adversarial attacks against our defense using 2 TITAN X GPUs. Table 17 shows the computational time required to break our defense when the adversary has varying amounts of information about our defense framework. In Table 17 we attack the defense framework by creating adversarial attacks against the adversary's substitute CNN and transfer the attacks to our defense framework [5], [104]. From Table 17 we can see that when the adversary has no knowledge

TABLE 17: Computation time required for breaking our defense framework on the CIFAR-10 dataset using the IFGSM and PGD attack with $\epsilon = 0.1$.

Bayesian CNN	MagNet	Pixel Defend	ShieldNets	Time to create 50 attacks**
✗	✗	✗	✗	38/34 hours
✗	✓	✗	✗	19/17 hours
✗	✗	✓	✗	28/25 hours
✗	✗	✗	✓	14/14 hours
✗	✓	✓	✗	9.55/6.30 hours
✗	✗	✓	✓	8.70/8.25 hours
✗	✓	✗	✓	6.65/5.40 hours
✓	✗	✗	✗	17/10 minutes
✓*	✗	✗	✗	34/31 hours
✓	✓	✓	✓	7.5/6 minutes

✓ is the part of the defense the adversary has knowledge of, whereas ✗ is the part of the defense protected from the adversary.
*We force all images to have one forced iteration of purification.
** Time to create 50 attacks is represented as x/y where x and y are the time taken to create 50 IFGSM and PGD attacks, respectively.

about our defense framework, it takes 38 hours to create 50 successful IFGSM adversarial attacks. To put this into perspective, it took 10 hours for Song *et al*. [27] to create 100 IFGSM attacks against their defense using 1 TITAN X GPU and 27 hours for Theagarajan *et al*. [28] to create 50 IFGSM attacks against their defense using 2 TITAN X GPUs.

4.7.1 Attacking the Bayesian CNN

In our defense the Bayesian CNN decides if an incoming image is adversarial or not before (i) passing it to the Black box classifier or (ii) rejecting the image. Hence, a natural target for an adversary to beat our defense would be to adversarially attack the Bayesian CNN. The optimization function for creating adversarial examples against the Bayesian CNN is shown in Eq. (18).

$$\arg\max \ ||\delta|| < \epsilon \tag{18a}$$
$$\textbf{S.T.} \ Y' \neq Y \tag{18b}$$
$$Aleatoric \ uncertainty < T_1 \tag{18c}$$
$$Epistemic \ uncertainty < T_2 \tag{18d}$$

As shown in Table 17 attacking the Bayesian CNN is by far the weakest point in our defense which can be exploited by an adversary provided the adversary has this information. But, optimizing Eq. (18) makes the adversarial images to be very close to the boundary of the original images in the input space thus resulting in a weakly perturbed adversarial image. We noticed that when we pass these weakly perturbed adversarial images through our ensemble of defense the resulting image is purified in just one iteration of purification. Hence, in Table 17 when we force all input images (regardless if they are adversarial or not) to at least one iteration of purification, these weakly perturbed adversarial examples are no longer adversarial and are correctly classified by the black box classifier. By doing so, even if the adversary has full knowledge about the Bayesian CNN but no information about our ensemble of defenses, it takes approximately 34 hours to break our defense compared to just 17 minutes when there is no forced purification. Moreover, if the adversary has full knowledge of our defense it takes only 6 minutes to create 50 successful PGD attacks, but doing so would violate our assumptions described in Section 3.1 (i.e. *security through obscurity*). This finding is corroborated by the works done in Ilyas *et al*. [107], where it is shown that once the adversary has information about the defense (such as model architecture, parameters, gradients, etc.), it is relatively easy to break the defense.

4.7.2 Attacking the Ensemble of Iterative Adversarial Defenses

From Table 17 we can see that when the adversary has partial information about our ensemble of defenses, the time taken to break the defense ranges from 28 to 6.65 hours. The reason for this is that with the inclusion of probabilistic generative networks, such as PixelDefend [27] and ShieldNets [28], the ensemble changes from being a deterministic system to a probabilistic system and in order to attack a probabilistic system, one needs to solve the stochastic gradient descent. The convergence rate for solving this is of the order of $O(1/\lambda)$, where λ is the convergence error and this is exponentially slower than the deterministic case. Additionally, the individual defenses are all trained independently and have independent parameters, hence a perturbation in a certain direction leading to a misclassification against a particular defense does not necessarily lead to a similar perturbation in the same direction for the other defenses within the ensemble. Moreover, breaking our defense requires a significant amount of probing and querying from the adversary's side and this can be limited by setting a threshold beyond which the adversary cannot probe the defense for a certain amount of time [107], hence further increasing the computation overhead.

5 Conclusions

This paper presented a novel privacy preserving adversarial defense framework for defending black box classifiers. The proposed framework has the ability to convert an existing single-step black box defense into an iterative defense and experimental results showed that using an ensemble of defenses outperforms the state-of-the-art. We demonstrated the relationship between an adversarial image and its corresponding purified image and proved the existence of a lower bound in the input space beyond which an image cannot be further purified. This paper proposed three novel knowledge distillation approaches that exploit prior meta-information of the training datasets, while preserving the privacy of the black box classifiers. Experimental results showed that crowd sourced images that are available in the public domain can be used to effectively distill the knowledge of the Black box classifier and still achieve reasonable performance in defending against adversarial attacks. Most importantly, neither our defense required any information about the parameters of the black box classifiers nor its training data, thus it preserved the privacy and significantly increased its adversarial robustness.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks", *arXiv preprint arXiv:1312.6199*, 2013.

[2] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale", *arXiv preprint arXiv:1611.01236*, 2016.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples", *arXiv preprint arXiv:1412.6572*, 2014.

[4] N. Carlini, and D. Wagner, "Towards evaluating the robustness of Neural Networks", *arXiv preprint arXiv:1608.04644*, 2016.

[5] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," *In Proceedings of the ACM Asia Conference on Computer and Communications Security*, pp. 506-519, 2017.

[6] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," *arXiv preprint arXiv:1511.03034*, 2015.

[7] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *In International Conference on Learning Representations*, 2018.

[8] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Painless adversarial training using maximal principle," *arXiv preprint arXiv:1905.00877*, 2019.

[9] N. Das, M. Shanbhogue, S. T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression," *arXiv preprint arXiv:1705.02900*, 2017.

[10] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of JPG compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[11] C. Guo, M. Rana, M. Cisse, and L. V. D. Maaten, "Countering adversarial images using input transformations," *International Conference on Learning Representations*, 2018.

[12] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Barrage of random transforms for adversarially robust defense," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6528–6537, 2019.

[13] H. Hosseini, Y. Chen, S. Kannan, B. Zhang, and R. Poovendran, "Blocking transferability of adversarial examples in black-box learning systems," *arXiv preprint arXiv:1703.04318*, 2017.

[14] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, 2018.

[15] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. "Distillation as a defense to adversarial perturbations against deep Neural Networks," *IEEE Symposium on Security and Privacy*, 2016.

[16] C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019.

[17] A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao, "Adversarial defense by restricting the hidden space of deep neural networks," IEEE International Conference on Computer Vision, pp. 3385-3394, 2019.

[18] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," *International Symposium on Research in Attacks, Intrusions, and Defenses*, 2018.

[19] G. S. Dhillon, K. Azizzadenesheli, Z. C. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," *International Conference on Learning Representations*, 2018.

[20] S. Ye, K. Xu, S. Liu, H. Cheng, J. H. Lambrechts, H. Zhang, A. Zhou, K. Ma, Y. Wang, and X. Lin, "Adversarial robustness Vs. model compression, or both," *IEEE International Conference on Computer Vision*, 2019.

[21] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," *International Conference on Learning Representations*, 2018.

[22] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," *arXiv preprint arXiv:1707.05373*, 2017.

[23] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[24] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," IEEE International Conference on Computer Vision, 2017.

[25] Q. Wang, W. Guo, K. Zhang, I. I. Ororbia, G. Alexander, X. Xing, X. Liu, and C. L. Giles, "Learning adversary-resistant deep neural networks," *arXiv preprint arXiv:1612.01401*, 2016.

[26] D. Meng, and H. Chen, "MagNet: A two-pronged defense against adversarial examples," *ACM SIGSAC Conference on Computer and Communications Security*, pp. 135–147, 2017.

[27] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "PixelDefend: Leveraging generative models to understand and defend against adversarial examples," *International Conference on Learning Representations*, 2018.

[28] R. Theagarajan, M. Chen, B. Bhanu, and J. Zhang, "Shieldnets: Defending against adversarial attacks using probabilistic adversarial robustness," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6988–6996, 2019.

[29] J. Lu, T. Issaranon, and D. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," *IEEE International Conference on Computer Vision*, pp. 446–454, 2017.

[30] Y. Gal, and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," *International Conference on Machine Learning*, pp. 1050–1059, 2016.

[31] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[32] Y. C. Lin, M. Y. Liu, M. Sun, and J. B. Huang, "Detecting adversarial attacks on neural network policies with visual foresight," *arXiv preprint arXiv:1710.00814*, 2017.

[33] J. Lu, T. Issaranon, and D. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," *IEEE International Conference on Computer Vision*, pp. 446–454, 2017.

[34] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[35] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[36] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P. Chen, Y. Wang, and X. Lin, "Evading real-time person detectors by adversarial t-shirt," *arXiv preprint arXiv:1910.11099*, 2019.

[37] S. T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "Shapeshifter: Robust physical adversarial attack on faster R-CNN object detector," *European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 52-68, 2018.

[38] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *International Conference on Learning Representations*, 2017.

[39] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems* 30(9), pp. 2805-2824, 2019.

[40] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, 23(5), pp.828-841, 2019.

[41] V. Khrulkov, and I. Oseledets, "Art of singular vectors and universal adversarial perturbations," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8562-8570, 2018.

[42] P. C. Van Oorschot, "Revisiting software protection," *International Conference on Information Security*, pp. 1-13, 2013.

[43] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," *ACM SIGSAC Conference on Computer and Communications Security*, pp. 308-318, 2016.

[44] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: an application of human behavior prediction," *AAAI Conference on Artificial Intelligence*, 30(1), 2016.

[45] M. A. P. Chamikara, P. Bertok, I. Khalil, D. Liu, and S. Camtepe, "privacy preserving face recognition utilizing differential privacy," *arXiv preprint arXiv:2005.10486*, 2020.

[46] C. Dwork, and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, 9(3-4), pp. 211-407, 2014.

[47] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," *IEEE Symposium on Security and Privacy*, pp. 3-18, 2017.

[48] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

[49] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," *ACM SIGSAC Conference on Computer and Communications Security*, pp. 259-274, 2019.

[50] N. Carlini, S. Deng, S. Garg, S. Jha, S. Mahloujifar, M. Mahmoody, S. Song, A. Thakurta, and F. Tramer, "An Attack on InstaHide: Is Private Learning Possible with Instance Encoding?," *arXiv preprint arXiv:2011.05315*, 2020.

[51] https://www.washingtonpost.com/technology/2019/09/12/california-could-become-largest-state-ban-facial-recognition-body-cameras/

[52] X. X. Zhu, D. Tuia, L. Mou, G. S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, 5(4), pp. 8-36, 2017.

[53] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, 37(3), pp. 362-386, 2020.

[54] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," *Neural Information Processing Systems*, pp. 742-751, 2017.

[55] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again Neural Networks," *International Conference on Machine Learning*, 2018.

[56] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff Nets: Stealing functionality of black-box models," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4954–4963, 2019.

[57] S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICARL: Incremental classifier and representation learning," *IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001-2010, 2017.

[58] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, 518(7540), pp.529-533, 2015.

[59] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1m: A dataset and benchmark for large-scale face recognition," *European Conference Computer Vision*, pages 87–102, 2016.

[60] Z. Luo, F. Branchaud-Charron, C. Lemaire, J. Konrad, S. Li, A. Mishra, A. Achkar, J. Eichel, and P. M. Jodoin, "MIO-TCD: A new benchmark dataset for vehicle classification and localization," *IEEE Transactions on Image Processing*, 27(10), pp.5129-5141, 2018.

[61] N. Das, M. Shanbhogue, S. T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression," *arXiv preprint arXiv:1705.02900*, 2017.

[62] G. Goswami, A. Agarwal, N. Ratha, R. Singh, and M. Vatsa, "Detecting and mitigating adversarial perturbations for robust face recognition," *International Journal of Computer Vision*, 127(6-7), pp. 719–742, 2019.

[63] N. Frosst, and G. Hinton, "Distilling a neural network into a soft decision tree," *arXiv preprint arXiv:1711.09784*, 2017.

[64] J. Ba, and R. Caruana, "Do deep nets really need to be deep?," *Neural Information Processing Systems*, pp. 2654–2662, 2014.

[65] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Neural Information Processing Systems*, pp. 2994–3003, 2017.

[66] S. Tan, R. Caruana, G. Hooker, and Y. Lou, "Distill-and-compare: Auditing black-box models using transparent model distillation," *AAAI/ACM Conference on AI, Ethics, and Society*, pp. 303-310, 2018.

[67] J. Wang, L. Gou, W. Zhang, H. Yang, and H. W. Shen, "Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation," *IEEE transactions on visualization and computer graphics*, 25(6), pp. 2168-2180, 2019.

[68] X. Song, F. Feng, X. Han, X. Yang, W. Liu, and L. Nie, "Neural compatibility modeling with attentive knowledge distillation," *International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 5-14, 2018.

[69] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor, "A deep hierarchical approach to lifelong learning in minecraft," *AAAI Conference on Artificial Intelligence*, 2017.

[70] E. J. Crowley, G. Gray, and A. J. Storkey, "Moonshine: Distilling with cheap convolutions," *Neural Information Processing Systems*, pp. 2888-2898, 2018.

[71] R. Theagarajan, F. Pala, and B. Bhanu, "EDeN: Ensemble of deep networks for vehicle classification," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 33-40, 2017.

[72] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[73] A. Rawat, M. Wistuba, and M. I. Nicolae, "Adversarial phenomenon in the eyes of Bayesian deep learning," *arXiv preprint arXiv:1711.08244*, 2017.

[74] Y. Gal, and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," *International Conference on Machine Learning*, pp. 1050-1059, 2016.

[75] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *International Conference on Machine Learning*, 2015.

[76] S. Kullback, and R. A. Leible, "On information and sufficiency," *The annals of mathematical statistics*, 22(1), pp.79-86, 1951.

[77] K. Shridhar, F. Laumann, and M. Liwicki, "Uncertainty estimations by softplus normalization in Bayesian convolutional neural networks with variational inference", *arXiv preprint arXiv:1806.05978*, 2018.

[78] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv cs.LG/1708.07747*, 2017.

[79] A. Krizhevsky, and G. Hinton. "Learning multiple layers of features from tiny images", *Technical report, Citeseer*, 2009.

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

[81] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," *International Conference on Machine Learning*, 2018.

[82] S. Jenni, and P. Favaro, "Deep bilevel learning," *European Conference on Computer Vision*, pp. 618-633, 2018.

[83] J. M. Köhler, M. Autenrieth, and W. H. Beluch, "Uncertainty based detection and relabeling of noisy image labels," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 33-37, 2019.

[84] L. Yao, and J. Miller, "Tiny imagenet classification with convolutional neural networks," *CS 231N*, 2(5), pp. 8, 2015.

[85] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, 115(3), pp. 211-252, 2015.

[86] C. Fellbaum, "WordNet," *The encyclopedia of applied linguistics*, 2012.

[87] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, 1(4), 541-551, 1989.

[88] J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 60–68, 2017.

[89] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 212–220, 2017.

[90] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy, "The devil of face recognition is in the noise," *European Conference on Computer Vision*, pp. 765–780, 2018.

[91] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," *European Conference on Computer Vision*, pp. 499–515, 2016.

[92] S. Fortunato, "Community detection in graphs," *Physics reports*, 486(3-5), pp. 75–174, 2010

[93] C. Jin, R. Jin, K. Chen, and Y. Dou, "A community detection approach to cleaning extremely large face database," *Computational Intelligence and Neuroscience*, 2018.

[94] R. Theagarajan, and B. Bhanu, "Defending black box facial recognition classifiers against adversarial attacks," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

[95] J. Wang, W. Bao, L. Sun, X. Zhu, B. Cao, and S. Y. Philip, "Private model compression via knowledge distillation," *AAAI Conference on Artificial Intelligence*, 33, pp. 1190-1197, 2019.

[96] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4133-4141, 2017.

[97] A. Mishra, and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," *International Conference on Learning Representations*, 2018.

[98] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282*, 2017.

[99] N. Narodytska, and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep Neural Networks," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1310-1318, 2017.

[100] A. Madry, A. Makelov, L. Schmidt,D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *International Conference on Learning Representations*, 2018.

[101] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," *International Conference on Learning Representations*, 2018.

[102] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," *arXiv preprint arXiv:1905.07121*, 2019.

[103] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9185-9193, 2018.

[104] N. Narodytska, and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep Neural Networks," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1310–1318, 2017.

[105] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[106] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[107] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," *International Conference on Machine Learning*, 2018.

[108] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," *International Conference on Machine Learning*, pp. 3866-3876, 2019.

[109] J. Uesato, B. O'donoghue, P. Kohli, and A. Oord, "Adversarial risk and the dangers of evaluating against weak attacks," *International Conference on Machine Learning*, pp. 5025-5034, 2018.

[110] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *International Conference on Machine Learning*, 2018.

[111] P. Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," *ACM Workshop on Artificial Intelligence and Security*, pp. 15-26, 2017.

[112] C. Vaccari, and A. Chadwick, "Deepfakes and disinformation: exploring the impact of synthetic political video on deception, uncertainty, and trust in news," *Social Media+ Society*, 6(1), 2020.

[113] P. Korshunov, and S. Marcel, "Vulnerability assessment and detection of deepfake videos," *IEEE International Conference on Biometrics*, pp. 1-6, 2019.

[114] N. Carlini, and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," *ACM Workshop on Artificial Intelligence and Security*, pp. 3-14, 2017.

[115] B. Bhanu, and V. Govindaraju (Eds.), "Multibiometrics for human identification," *Cambridge University Press*, 2010.

Rajkumar Theagarajan (S'14) received the B.E degree in electronics and communication engineering from the Anna University, Chennai, India, in 2014 and M.S. and Ph.D. degree in electrical and computer engineering from the University of California, Riverside, CA, USA, in 2016 and 2020, respectively. Currently he is with KLA Corporation. His research interests include computer vision and machine learning.

Bir Bhanu (F'95, LF'17) received B.S. (with Hons.) from IIT-BHU; M.E (with Distinction) from BITS (Pilani); S.M. and E.E. in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA; Ph.D. in electrical engineering from the University of Southern California, Los Angeles, CA and M.B.A. from the University of California, Irvine, CA. He is the Bourns endowed University of California Presidential Chair in Engineering, the Distinguished Professor of electrical and computer engineering and the Founding Director of the interdisciplinary Center for Research in Intelligent Systems (1998-2019) and the Visualization and Intelligent Systems Laboratory (1991-) at the University of California, Riverside (UCR), CA. He is the Founding Professor of electrical engineering with UCR and served as its first Chair (1991-94). He has been the cooperative Professor of computer science and engineering (since 1991), bioengineering (since 2006) and mechanical engineering (since 2008). Recently he served as the Interim Chair of the Department of Bioengineering from 2014-16. He also served as the Director of the National Science Foundation graduate research and training program in video bioinformatics with UCR. Prior to joining UCR in 1991, he was a Senior Honeywell Fellow with Honeywell Inc. He has published extensively and has 18 patents. His research interests include computer vision, pattern recognition and data mining, machine learning, artificial intelligence, image processing, image and video database, graphics and visualization, robotics, human-computer interactions, and biological, medical, military and intelligence applications. Dr. Bhanu is a Fellow of IEEE, AAAS, IAPR, SPIE, NAI and AIMBE.