Toward Adversarial Robustness by Diversity in an Ensemble of Specialized Deep Neural Networks

Mahdieh Abbasi[1(✉)], Arezoo Rajabi[2], Christian Gagné[1,3], and Rakesh B. Bobba[2]

[1] IID, Université Laval, Québec, Canada
mahdieh.abbasi.1@ulaval.ca
[2] Oregon State University, Corvallis, USA
[3] Mila, Canada CIFAR AI Chair, Quebec city, Canada

Abstract. We aim at demonstrating the influence of diversity in the ensemble of CNNs on the detection of black-box adversarial instances and hardening the generation of white-box adversarial attacks. To this end, we propose an ensemble of diverse specialized CNNs along with a simple voting mechanism. The diversity in this ensemble creates a gap between the predictive confidences of adversaries and those of clean samples, making adversaries detectable. We then analyze how diversity in such an ensemble of specialists may mitigate the risk of the black-box and white-box adversarial examples. Using MNIST and CIFAR-10, we empirically verify the ability of our ensemble to detect a large portion of well-known black-box adversarial examples, which leads to a significant reduction in the risk rate of adversaries, at the expense of a small increase in the risk rate of clean samples. Moreover, we show that the success rate of generating white-box attacks by our ensemble is remarkably decreased compared to a vanilla CNN and an ensemble of vanilla CNNs, highlighting the beneficial role of diversity in the ensemble for developing more robust models.

1 Introduction

Convolutional Neural Networks (CNNs) are now a common tool in many computer vision tasks with a great potential for deployement in real-world applications. Unfortunately, CNNs are strongly vulnerable to minor and imperceptible adversarial modifications of input images a.k.a. adversarial examples or adversaries. In other words, generalization performance of CNNs can be significantly dropped in the presence of adversaries. While identifying such benign-looking adversaries from their appearance is not always possible for human observers, distinguishing them from their predictive confidences by CNNs is also challenging since these networks, as uncalibrated learning models [1], misclassify them with high confidence. Therefore, the lack of robustness of CNNs to adversaries can lead to significant issues in many security-sensitive real-world applications such as self-driving cars [2].

C. Goutte and X. Zhu (Eds.): Canadian AI 2020, LNAI 12109, pp. 1–14, 2020.
https://doi.org/10.1007/978-3-030-47358-7_1

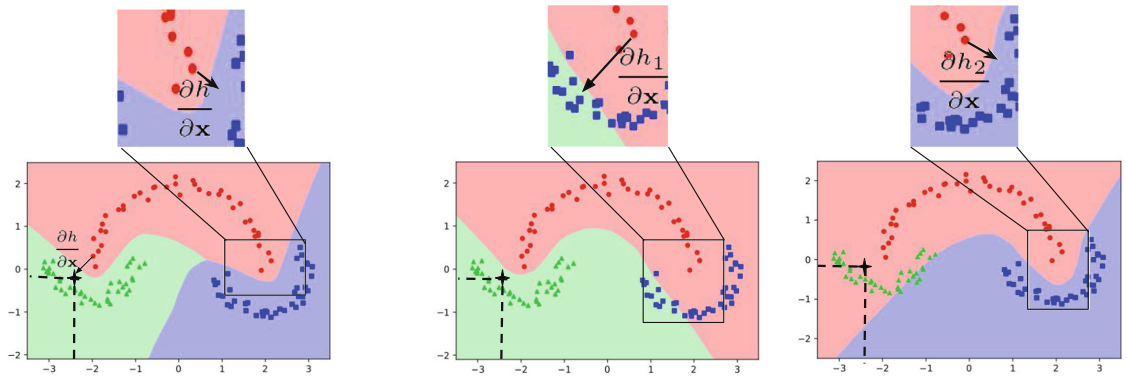


Fig. 1. A schematic explanation of ensemble of specialists for a 3-classes classification. On the left, a generalist ($h(.)$) trained on all 3 classes. In the middle and on the right, two specialist binary-classifiers $h_1(.)$ and $h_2(.)$ are trained on different subsets of classes, i.e. respectively (red,green) and (red, blue). A black-box attack, shown by **a black star**, which fools a generalist classifier (left), can be classified as different classes by the specialists, creating diversity in their predictions. Moreover, generation of a white-box adversarial example by the specialists can create two different fooling directions toward two unlike fooling classes. The fooling directions (in term of derivatives) are shown by black arrows in zoomed-in figures. Such different fooling directions by the specialists can harden the generation of high confidence white-box attacks (Sect. 3). *Thus, by leveraging diversity in an ensemble of specialists, without the need of adversarial training, we may mitigate the risk of adversarial examples.* (Color figure online)

To address this issue, one line of thought, known as *adversarial training*, aims at enabling CNNs to *correctly classify any type of adversarial examples* by augmenting a clean training set with a set of adversaries [3–7]. Another line of thought is to devise detectors to discriminate adversaries from their clean counterparts by training the detectors on a set of clean samples and their adversarials ones [4,8–10]. However, the performance of these approaches, by either increasing correct classification or detecting adversaries, is highly dependent on accessing a holistic set containing various types of adversarial examples. Not only generating such a large number of adversaries is computationally expensive and impossible to be made exhaustively, but adversarial training does not necessarily grant robustness to unknown or unseen adversaries [11,12].

In this paper, we aim at detecting adversarial examples by predicting them with high uncertainty (low confidence) through leveraging diversity in an ensemble of CNNs, without requiring a form of adversarial training. To build a diverse ensemble, we propose forming a *specialists ensemble*, where each specialist is responsible for classifying a different subset of classes. The specialists are defined so as to encourage divergent predictions in the presence of adversarial examples, while making consistent predictions for clean samples (Fig. 1). We also devise a simple voting mechanism to merge the specialists' predictions to efficiently compute the final predictions. As a result of our method, we are enforcing a gap

between the predictive confidences of adversaries (i.e., low confidence predictions) and those of clean samples (i.e., high confidence predictions). By setting a threshold on the prediction confidences, we can expect to properly identify the adversaries. Interestingly, we provably show that the predictive confidence of our method in the presence of disagreement (high entropy) in the ensemble is upper-bounded by $0.5 + \epsilon'$, allowing us to have a global fixed threshold (i.e., $\tau = 0.5$) without requiring fine-tuning of the threshold. Moreover, we analyze our approach against the black-box and white-box attacks to demonstrate how, without adversarial training and only by diversity in the ensemble, one may design more robust CNN-based classification systems. The contributions of our paper are as follows:

- We propose an ensemble of diverse specialists along with a simple and computationally efficient voting mechanism in order to predict the adversarial examples with low confidence while keeping the predictive confidence of the clean samples high, without training on any adversarial examples.
- In the presence of high entropy (disagreement) in our ensemble, we show that the maximum predictive confidence can be upper-bounded by $0.5 + \epsilon'$, allowing us to use a fixed global detection threshold of $\tau = 0.5$.
- We empirically exhibit that several types of black-box attacks can be effectively detected with our proposal due to their low predictive confidence (i.e., ≤ 0.5). Also, we show that attack-success rate for generating white-box adversarial examples using the ensemble of specialists is considerably lower than those of a single generalist CNN and a ensemble of generalists (a.k.a pure ensemble).

2 Specialists Ensemble

Background: For a K-classification problem, let us consider training set of $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ with $\mathbf{x}_i \in \mathcal{X}$ as an input sample along with its associated ground-truth class k, shown by a one-hot binary vector $\mathbf{y}_i \in [0,1]^K$ with a single 1 at its k-th element. A CNN, denoted by $h_{\mathcal{W}} : \mathcal{X} \to [0,1]^K$, maps a given input to its conditional probabilities over K classes. The classifier $h_{\mathcal{W}}(\cdot)$[1] is commonly trained through a cross-entropy loss function minimization as follows:

$$\min_{\mathcal{W}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(h(\mathbf{x}_i), \mathbf{y}_i; \mathcal{W}) = -\frac{1}{N} \sum_{i=1}^{N} \log h_{k^*}(\mathbf{x}_i), \tag{1}$$

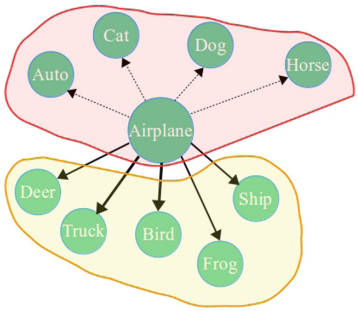
where $h_{k^*}(\mathbf{x}_i)$ indicates the estimated probability of class k^* corresponding to the true class of given sample $\mathbf{x}_i$. At the inference time, the threshold-based approaches like our approach define a threshold τ in order to reject the instances with lower predictive confidence than τ as an extra class $K+1$:

$$d(\mathbf{x}|\tau) = \begin{cases} \text{argmax}_k \, h_k(\mathbf{x}), & \text{if } \max_k h_k(\mathbf{x}) > \tau \\ K+1, & \text{otherwise} \end{cases}. \tag{2}$$

[1] For convenience, $\mathcal{W}$ is dropped from $h_{\mathcal{W}}(\cdot)$.

	Airplane	Auto	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Airplane	0.00	1.60	22.00	3.88	15.20	0.22	18.46	1.70	17.70	19.24
Auto	2.70	0.00	7.20	1.94	5.46	0.06	7.86	0.90	8.50	65.38
Bird	3.70	0.58	0.00	13.34	32.24	0.98	33.02	2.78	3.28	10.08
Cat	0.82	0.30	12.58	0.00	20.92	6.86	43.38	6.24	1.46	7.44
Deer	0.90	0.24	21.00	7.30	0.00	0.58	51.94	8.94	0.94	8.16
Dog	0.66	0.26	16.94	14.92	15.88	0.00	36.96	5.86	1.74	6.78
Frog	0.60	0.76	28.02	16.94	41.14	0.52	0.00	3.48	0.74	7.80
Horse	0.92	0.36	6.40	5.66	32.44	1.62	47.16	0.00	0.98	4.46
Ship	11.10	4.12	20.94	4.42	7.12	0.10	14.28	0.84	0.00	37.08
Truck	3.88	23.92	7.24	2.64	13.44	0.22	42.06	1.74	4.86	0.00

(a) CIFAR-10 FGS fooling matrix



(b) The expertise domains of "Airplane" class

Fig. 2. (a) Fooling matrix of FGS adversaries for CIFAR-10, which is computed from 5000 randomly selected FGS adversaries (500 per class). Each row shows the fooling rates (in percentage) from a true class to other classes (rows and columns are true and fooling classes, respectively). (b) An example of forming expertise domains for class "Airplane": its high likely fooled classes (in yellow zone) and less likely fooled classes (in red zone) are forming two expertise domains. (Color figure online)

2.1 Ensemble Construction

We define the expertise domain of the specialists (i.e. the subsets of classes) by separating each class from its most likely fooled classes. We later show in Sect. 3 how separation of each class from its high likely fooling classes can promote entropy in the ensemble, which in turns leads to predicting adversaries with low confidence (high uncertainty).

To separate the most fooling classes from each other, we opt to use the fooling matrix of FGS adversarial examples $\mathbf{C} \in \mathbb{R}^{K \times K}$. This matrix reveals that the clean samples from each true class have a high tendency to being fooled toward a limited number of classes not uniformly toward all of them (Fig. 2(a)). The selection of FGS adversaries is two-fold; their generation is computationally inexpensive, and they are highly transferable to many other classifiers, meaning that different classifiers (e.g. with different structures) behave in similar manner in their presence, i.e. fooled to the same classes [13–15].

Using each row of the fooling matrix (i.e. $\mathbf{c}_i$), we define two expertise domains for i-th true class so as to split its high likely fooling classes from its less likely fooling classes as follows (Fig. 2(b)):

- Subset of high likely fooling classes of i: $\mathbb{U}_i = \cup\{j\}$ if $c_{ij} > \mu_i$, $j \in \{1, \ldots, K\}$
- Subset of less likely fooling classes of i: $\mathbb{U}_{i+K} = \{1, \ldots, K\} \setminus \mathbb{U}_i$,

where $\mu_i = \sum_{j=1}^{K} c_{ij}$ (average of fooling rates of i-th true class). Repeating the above procedure for all K classes makes $2K$ subsets (expertise domains) for a K classification problem. Note that the duplicated expertise domains can be removed so as to avoid having multiple identical expertise domains (specialists).

Afterwards, for each expertise domain, one specialist is trained in order to form an ensemble of specialist CNNs. A generalist (vanilla) CNN, which trained

on the samples belonging to all classes, is also added to this ensemble. The ensemble involving $M \leq 2K+1$ members is represented by $\mathcal{H} = \{h^1, \ldots, h^M\}$, where $h^j(\cdot) \in [0,1]^K$ is j-th individual CNN mapping a given input to conditional probability over its expert classes, i.e. the probability of the classes out of its expertise domain is fixed to zero.

2.2 Voting Mechanism

To compute the final prediction out of our ensemble for a given sample, we need to activate relevant specialists, then averaging their prediction along with that of the generalist CNN. Note that we cannot simply use the generalist CNN to activate specialists since in the presence of adversaries it can be fooled, then causing selection (activation) of the wrong specialists. In Algorithm 1, we devise a simple and computationally efficient voting mechanism to activate those relevant specialists, then averaging their predictions.

Algorithm 1. Voting Mechanism

Input: Ensemble $\mathcal{H} = \{h^1, \ldots, h^M\}$, expertise domains $\mathbb{U} = \{\mathbb{U}_1, \ldots, \mathbb{U}_M\}$, input $\mathbf{x}$
Output: Final prediction $\bar{h}(\mathbf{x}) \in [0,1]^K$

1: $v_k(\mathbf{x}) \leftarrow \sum_{j=1}^{M} \mathbb{I}\left(k = \text{argmax}_{i=1}^{K} h_i^j(\mathbf{x})\right),\ k = 1, \ldots, K$
2: $k^* \leftarrow \text{argmax}_{k=1}^{K}\, v_k(\mathbf{x})$
3: **if** $v_{k^*}(\mathbf{x}) = \lceil \frac{M}{2} \rceil$
4: $\quad \mathcal{H}_{k^*} \leftarrow \{h^i \in \mathcal{H} \,|\, k^* \in \mathbb{U}_i\}$
5: $\quad \bar{h}(\mathbf{x}) \leftarrow \frac{1}{|\mathcal{H}_{k^*}|} \sum_{h^i \in \mathcal{H}_{k^*}} h^i(\mathbf{x})$
6: **else**
7: $\quad \bar{h}(\mathbf{x}) \leftarrow \frac{1}{M} \sum_{h^i \in \mathcal{H}} h^i(\mathbf{x})$
8: **return** $\bar{h}(\mathbf{x})$

Let us first introduce the following elements for each class i:

- The actual number of votes for i-th class by the ensemble for a given sample $\mathbf{x}$: $v_i(\mathbf{x}) = \sum_{j=1}^{M} \mathbb{I}\left(i = \text{argmax}_{\{1,\ldots K\}}\, h^j(\mathbf{x})\right)$, i.e. it shows the number of the members that classify $\mathbf{x}$ to i-th class.
- The maximum possible number of votes for i-th class is $\lceil \frac{M}{2} \rceil \leq K+1$. Recall that for each row, we split all K classes into two expertise domains, where class i is included in one of them. Considering all K rows and the generalist, we end up having at maximum $K+1$ subsets that involve class i.

As described in Algorithm 1, for a given sample $\mathbf{x}$, if there is a class with its actual number of votes equal to its expected number of votes, i.e. $v_i(\mathbf{x}) = \lceil \frac{M}{2} \rceil$, then it means all of the specialists, which are trained on i-th class, are simultaneously voting (classifying) for it. We call such a class *a winner class*. Then, the specialists CNNs voting to the winner class are activated to compute the final prediction (lines 3–5 of Algorithm 1), producing a certain prediction

(with high confidence). Note that in the presence of clean samples, the relevant specialists in the ensemble are expected to do agree on the true classes since they, as strong classifiers, have high generalization performance on their expertise domains.

If no class obtains its maximum expected number of votes (i.e. $\nexists i,\ v_i(\mathbf{x}) = \lceil \frac{M}{2} \rceil$), it means that the input $\mathbf{x}$ leads the specialists to disagree on a winner class. In this situation, when no agreement exists in the ensemble, all the members should be activated to compute the final prediction (line 7 of Algorithm 1). Averaging of the predictions by all the members leads to a final prediction with high entropy (i.e. low confidence). Indeed, a given sample that creates a disagreement (entropy) in the ensemble is either a hard-to-classify sample or an abnormal sample (e.g. adversarial examples).

Using the voting mechanism for this specialists ensemble, we can create a gap between the predictive confidences of clean samples (having high confidence) and those of adversaries (having low confidence). Finally, using a threshold τ on these predictive confidences, the unusual samples are identified and rejected. In the following, we argue that our voting mechanism enables us to set a global fixed threshold $\tau = 0.5$ to perform identification of adversaries. This is unlike some threshold-based approaches [10,16] that need to tune different thresholds for various datasets and their types of adversaries.

Corollary 1. *In a disagreement situation, the proposed voting mechanism makes the highest predictive confidence to be upper-bounded by* $0.5 + \epsilon'$ *with* $\epsilon' = \frac{1}{2M}$.

Proof. Consider a disagreement situation in the ensemble for a given $\mathbf{x}$, where all the members are averaged to create $\bar{h}(\mathbf{x}) = \frac{1}{M} \sum_{h^j \in \mathcal{H}} h^j(\mathbf{x})$. The highest predictive confidence of $\bar{h}(\mathbf{x})$ belongs to the class that has the largest number of votes, i.e. $m = \max[v_1(\mathbf{x}), \ldots, v_K(\mathbf{x})]$. Let us represent these m members that are voting to this class (k-th class) as $\mathcal{H}_k = \{h^j \in \mathcal{H} \,|\, k \in \mathbb{U}_j\}$. Since each individual CNNs in the ensemble are basically uncalibrated learners (having very high confident prediction for a class and near to zero for the remaining classes), the confidence probability of k-th class of those excluded members from $\mathcal{H}_k$ (those that do not vote for k-th class) can be negligible. Thus, their prediction can be simplified as $\bar{h}_k(\mathbf{x}) = \frac{1}{M} \sum_{h^j \in \mathcal{H}_k} h^j_k(\mathbf{x}) + \frac{\epsilon}{M} \approx \frac{1}{M} \sum_{h^j \in \mathcal{H}_k} h^j_k(\mathbf{x})$ (the small term $\frac{\epsilon}{M}$ is discarded). Then, from the following inequality $\sum_{h^j \in \mathcal{H}_k} h^j_k(\mathbf{x}) \leq m$, we have $\frac{1}{M} \sum_{h^j \in \mathcal{H}_k} h^j_k(\mathbf{x}) \leq \frac{m}{M}$ (I).

On the other hand, due to having no winner class, we know that $m < \lceil \frac{M}{2} \rceil$ (or $m < \frac{M}{2} + \frac{1}{2}$), such that by multiplying it by $\frac{1}{M}$ we obtain $\frac{m}{M} < \frac{1}{2} + \frac{1}{2M}$ (II).

Finally considering (I) and (II) together, it derives $\frac{1}{M} \sum_{h^j \in \mathcal{H}_k} h^j_k(\mathbf{x}) < 0.5 + \frac{1}{2M}$. For the ensemble with a large size, e.g. likewise our ensemble, the term $\epsilon' = \frac{1}{2M}$ is small. Therefore, it shows the class with the maximum probability (having the maximum votes) can be upper-bounded by $0.5 + \epsilon'$. ■

3 Analysis of Specialists Ensemble

Here, we first explain how adversarial examples give rise to entropy in our ensemble, leading to their low predictive confidence (with maximum confidence of $0.5+\epsilon'$). As well, we examine the role of diversity in our ensemble, which harden the generation of white-box adversaries.

In a **black-box** attack, we assume that the attacker is not aware of our ensemble of specialists, thus generates some adversaries from a pre-trained vanilla CNN $g(\cdot)$ to mislead our underlying ensemble. Taking a pair of an input sample with its true label, i.e. $(\mathbf{x}, k)$, an adversary $\mathbf{x}' = \mathbf{x} + \delta$ fools the model g such that $k = \text{argmax}\, g(\mathbf{x})$ while $k' = \text{argmax}\, g(\mathbf{x}')$ with $k' \neq k$, where k' is one of those most-likely fooling classes for class k (i.e. $k' \in \mathbb{U}_k$). Among the specialists that are expert on k, at least one of them does not have k' in their expertise domains since we intentionally separated k-th class from its most-likely fooling classes when defining its expertise domains (Sect. 2.1). Formally speaking, denote those expertise domains comprising class k as follows $\mathcal{U}^k = \{\mathbb{U}_j \,|\, k \in \mathbb{U}_j\}$ where (I) $\mathbb{U}_j \neq \mathbb{U}_i \;\forall \mathbb{U}_i,\; \mathbb{U}_j \in \mathcal{U}^k$ and (II) $k' \notin \cap\, \mathcal{U}^k$. Therefore, regarding the fact that (I) the expertise domains comprising k are different and (II) their shared classes do not contain k', it is not possible that all of their corresponding specialists models are fooled simultaneously toward k'. In fact, these specialists may vote (classify) differently, leading to a disagreement on the fooling class k'. So, due to this disagreement in the ensemble with no winner class, all the ensemble's members are activated, resulting in prediction with high uncertainty (low confidence) according to Corollary 1. Generally speaking, if $\{\cap\, \mathcal{U}^k\} \setminus k$ is a small or an empty set, harmoniously fooling the specialist models, which are expert on k, is harder.

In a **white-box attack**, an attacker attempts to generate adversaries to *confidently* fool the ensemble, meaning the adversaries should simultaneously activate *all* of the specialists that comprise the fooling class in their expertise domain. Otherwise, if at least one of these specialists is not fooled, then our voting mechanism results in adversaries with low confidence, which can then be automatically rejected using the threshold ($\tau = 0.5$). In the rest we bring some justifications on the hardness of generating high confidence gradient-based attacks from the specialists ensemble.

Instead of dealing with the gradient of one network, i.e. $\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}$, the attacker should deal with the gradient of the ensemble, i.e. $\frac{\partial \bar{h}(\mathbf{x})}{\partial \mathbf{x}}$, where $\bar{h}(\mathbf{x})$ computed by line 5 or line 7 of Algorithm. 1. Formally, to generate a gradient-based adversary from the ensemble for a given labeled clean input sample $(\mathbf{x}, \mathbf{y} = k)$, the derivative of the ensemble's loss, i.e. $\mathcal{L}(\bar{h}(\mathbf{x}), \mathbf{y}) = -\log \bar{h}_k(\mathbf{x})$, w.r.t. $\mathbf{x}$ is as follows:

$$\frac{\partial \mathcal{L}(\bar{h}(\mathbf{x}), \mathbf{y})}{\partial \mathbf{x}} = \frac{\partial \mathcal{L}}{\partial \bar{h}_k(\mathbf{x})} \frac{\partial \bar{h}_k(\mathbf{x})}{\partial \mathbf{x}} = \underbrace{-\frac{1}{\bar{h}_k(\mathbf{x})}}_{\beta} \frac{\partial \bar{h}_k(\mathbf{x})}{\partial \mathbf{x}} = \beta \frac{1}{|\mathcal{H}_k|} \sum_{h^i \in \mathcal{H}_k} \frac{\partial h^i_k(\mathbf{x})}{\partial \mathbf{x}}. \quad (3)$$

Initially $\mathcal{H}_k$ indicates the set of activated specialists voting for class k (true label) plus the generalist for the given input $\mathbf{x}$. Since the expertise domains of the activated specialists are different ($\mathcal{U}^k = \{\mathbb{U}_j \,|\, k \in \mathbb{U}_j\}$), most likely their derivative are diverse, i.e. fooling toward different classes, which in turn creates perturbations in various fooling directions (Fig. 1). Adding such diverse perturbation to a clean sample may promote disagreement in the ensemble, where no winner class can be agreed upon. In this situation, when all of the members are activated, the generated adversarial sample is predicted with a low confidence, thus can be identified. For the iterative attack algorithms, e.g. I-FGS, the process of generating adversaries may continue using the derivative of all of the members, adding even more diverse perturbations, which in turn makes reaching to an agreement in the ensemble on a winner fooling class even more difficult.

4 Experimentation

Evaluation Setting: Using MNIST and CIFAR-10, we investigate the performance of our method for reducing the risk rate of black-box attacks (Eq. 5) due to of their detection, and reducing the success rate of creating white-box adversaries. Two distinct CNN configurations are considered in our experimentation: for *MNIST*, a basic CNN with three convolution layers of respectively 32, 32, and 64 filters of 5×5, and a final fully connected (FC) layer with 10 output neurons. Each of these convolution layers is followed by a ReLU and 3×3 pooling filter with stride 2. For *CIFAR-10*, a VGG-style CNN (details in [17]) is used. For both CNNs, we use SGD with a Nesterov momentum of 0.9, L2 regularization with its hyper-parameter set to 10^{-4}, and dropout ($p = 0.5$) for the FC layers. For the evaluation purposes, we compare our ensemble of specialists with a vanilla (naive) CNN, and a pure ensemble, which involves 5 vanilla CNNs being different by random initialization of their parameters.

Evaluation Metrics: To evaluate a predictor $h(\cdot)$ that includes a rejection option, we report a risk rate $E_D|\tau$ on a clean test set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ at a given threshold τ, which computes the ratio of the (clean) samples that are *correctly classified but rejected* due to their confidence less than τ and those that are *misclassified but not rejected* due to a confidence value above τ:

$$E_D|\tau = \frac{1}{N}\sum_{i=1}^{N}\Big(\big(\mathbb{I}[d(\mathbf{x}_i|\tau) \neq K+1] \;\times\; \mathbb{I}[\operatorname{argmax} h(\mathbf{x}_i) \neq \mathbf{y}_i]\big) + \big(\mathbb{I}[d(\mathbf{x}_i|\tau) = K+1] \;\times\; \mathbb{I}[\operatorname{argmax} h(\mathbf{x}_i) = \mathbf{y}_i]\big)\Big). \tag{4}$$

In addition, we report the risk rate $E_A|\tau$ on each adversaries set, i.e. $\mathcal{A} = \{(\mathbf{x}'_i, \mathbf{y}_i)\}_{i=1}^{N'}$ including pairs of an adversarial example $\mathbf{x}'_i$ associated by its true label, to show the percentage of misclassified adversaries that are not rejected due to their confidence value above τ:

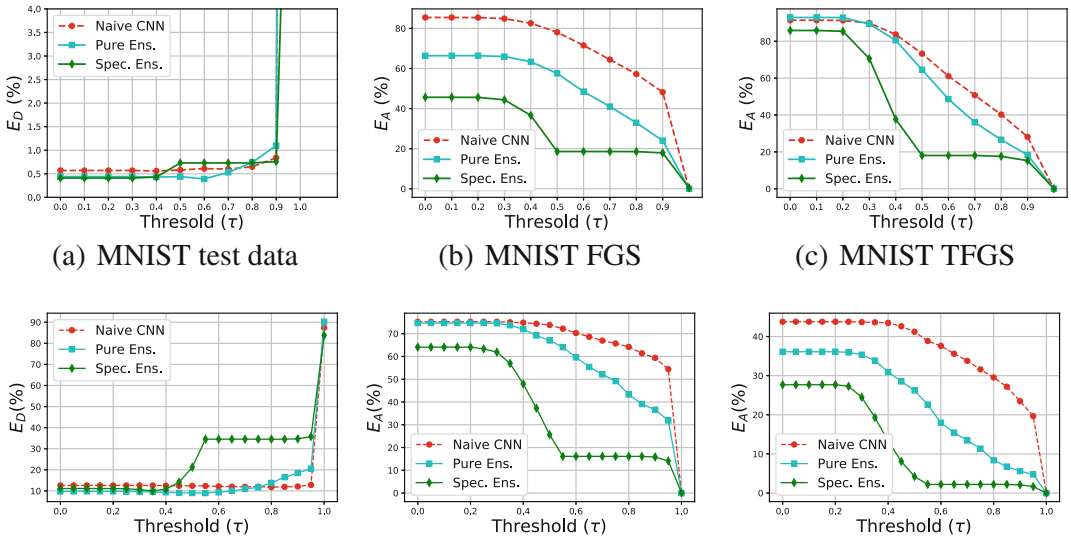
$$E_A|\tau = \frac{1}{N'}\sum_{i=1}^{N'}\big(\mathbb{I}[d(\mathbf{x}'_i|\tau) \neq K+1] \;\times\; \mathbb{I}[\operatorname{argmax} h(\mathbf{x}'_i) \neq \mathbf{y}_i]\big)\,. \tag{5}$$

4.1 Empirical Results

Black-box Attacks: To assess our method on different types of adversaries, we use various attack algorithms, namely FGS [5], TFGS [7], DeepFool (DF) [18], and CW [19]. To generate the black-box adversaries, we use another vanilla CNN, which is different from all its counterparts involved in the pure ensemble– by using different random initialization of its parameters. For FGS and T-FGS algorithms we generate 2000 adversaries with $\epsilon = 0.2$ and $\epsilon = 0.03$, respectively, for randomly selected clean test samples from MNIST and CIFAR-10. For CW attack, due to the high computational burden required, we generated 200 adversaries with $\kappa = 40$, where larger κ ensures generation of high confidence and highly transferable CW adversaries.

Figure 3 presents risk rates $(E_D|\tau)$ of different methods on clean test samples of MNIST (first row) and those of CIFAR-10 (second row), as well as their corresponding adversaries $E_A|\tau$, as functions of threshold (τ). As it can be seen from Fig. 3, by increasing the threshold, more adversaries can be detected (decreasing E_A) at the cost of increasing E_D, meaning an increase in the rejection of the clean samples that are correctly classified.

To appropriately compare the methods, we find an optimum threshold that creates small E_D and E_A collectively, i.e. $\text{argmin}_\tau\, E_D|\tau + E_A|\tau$. Recall that, as corollary 1 states, in our ensemble of specialists, we can fix the threshold of our ensemble to $\tau^* = 0.5$. In Table 1, we compare the risk rates of our ensemble with those of pure ensemble and vanilla CNN at their corresponding optimum thresholds. For MNIST, our ensemble outperforms naive CNN and pure ensemble as it detects a larger portion of MNIST adversaries while its risk rate on the clean samples is only marginally increased. Similarly, for CIFAR-10, our approach can detect a significant portion of adversaries at $\tau^* = 0.5$, reducing the risk rates



(a) MNIST test data (b) MNIST FGS (c) MNIST TFGS

(d) CIFAR-10 test data (e) CIFAR-10 FGS (f) CIFAR-10 TFGS

Fig. 3. The risk rates on the clean test samples and their black-box adversaries as the function of threshold (τ) on the predictive confidence.

Table 1. The risk rate of the clean test set ($E_D|\tau*$) along with that of black-box adversarial examples sets ($E_A|\tau*$) are shown in percentage at the optimum threshold of each method. The methods with the lowest collective risk rate (i.e. $E_A + E_D$) is underlined, while the best results for the two types of risk considered independently are in bold.

Task	Methods	Adversaries			
		FGS	TFGS	CW	DeepFool
		E_A / E_D	E_A / E_D	E_A / E_D	E_A / E_D
MNIST	Naive CNN	48.21 / 0.84	28.15 / 0.84	41.5 / 0.84	88.68 / 0.84
	Pure ensemble	24.02 / 1.1	18.35 / 1.1	28.5 / 1.1	72.73 / 1.1
	Specialists ensemble	**18.58 / 0.73**	**18.05 / 0.73**	**24 / 0.73**	**54.24 / 0.73**
CIFAR-10	Naive CNN	59.37 / **12.11**	23.47 / **12.11**	51.5 / **12.11**	28.81 / 12.11
	Pure ensemble	36.59 / 18.5	8.37 / 13.79	4.0 / 13.79	7.7 / 18.5
	Specialists ensemble	**25.66** / 21.25	**4.21** / 21.25	**3.5** / 21.25	6.02 / 21.25

on adversaries. However, at this threshold, our approach has higher risk rate on the clean samples than that of two other methods.

White-box Attacks: In the white-box setting, we assume that the attacker has full access to a victim model. Using each method (i.e. naive CNN, pure ensemble, and specialists ensemble) as a victim model, we generate different sets of adversaries (i.e. FGS, Iterative FGS (I-FGS), and T-FGS). A successful adversarial attack $\mathbf{x}'$ is achieved once the underlying model misclassifies it with a confidence higher than its optimum threshold τ^*. When the confidence for an adversarial example is lower than $\tau*$, it can be easily detected (rejected), thus it is not considered as a successful attack.

We evaluate the methods by their *white-box attacks success rates*, indicating the number of successful adversaries that satisfies the aforementioned conditions (i.e. a misclassification with a confidence higher than τ^*) during t iterations of the attack algorithm. Table 2 exhibits the success rates of white-box adversaries (along with their used hyper-parameters) generated by naive CNN ($\tau^* = 0.9$),

Table 2. Success rate of white-box adversarial examples (lower is better) generated by naive CNN, pure ensemble (5 generalists), and specialists ensemble at their corresponding optimum threshold. An successful white-box adversarial attack should fool the underlying model with a confidence higher than its optimum τ^*.

	MNIST			CIFAR-10		
Methods	Adversaries					
	FGS	T-FGS	I-FGS	FGS	T-FGS	I-FGS
	ϵ=0.2	ϵ=0.2	ϵ=2×10^{-2}	ϵ=3×10^{-2}	ϵ=3×10^{-2}	ϵ=3×10^{-3}
Naive CNN	89.94	66.16	66.84	86.16	81.38	93.93
Pure ensemble	71.58	50.64	48.62	42.65	13.96	45.78
Specialists ensemble	**45.15**	**27.43**	**13.63**	**34.1**	**7.43**	**34.20**

pure ensemble ($\tau^* = 0.9$), and specialists ensemble ($\tau^* = 0.5$). For the benchmark datasets, the number of iterations of FGS and T-FGS is 2 while that of iterative FGS is 10. As it can be seen in Table 2, the success rates of adversarial attacks using ensemble-based methods are smaller than those of naive CNN since diversity in these ensembles hinders generation of adversaries with high confidence.

Gray-box CW Attack: In the gray-box setting, it is often assumed that the attacker is aware of the underlying defense mechanism (e.g. specialists ensemble in our case) but has no access to its parameters and hyper-parameters. Following [20], we evaluate our ensemble on CW adversaries generated by another specialists ensemble, composed of 20 specialists and 1 generalist for 100 randomly selected MNIST samples. Evaluation of our specialists ensemble on these targeted gray-box adversaries (called "gray-box CW") reveals that our ensemble provides low confidence predictions (i.e. lower than 0.5) for 74% of them (thus able to reject them) while 26% have confidence more than 0.5 (i.e. non-rejected adversaries). Looking closely at those non-rejected adversaries in Fig. 4, it can be observed that some of them can even mislead a human observer due to adding very visible perturbation, where the appearance of digits are significantly distorted.

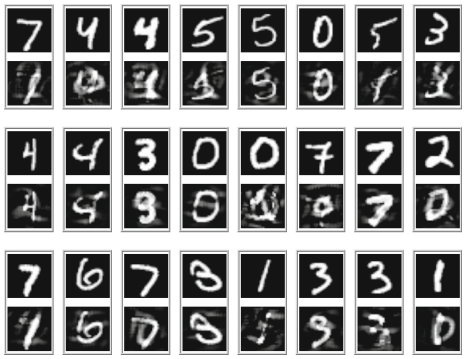


Fig. 4. Gray-box CW adversaries that confidently fool our specialists ensemble. According to the definition of adversarial example, however, some of them are not actually adversaries due to the significant visual perturbations.

5 Related Works

To address the issue of robustness of deep neural networks, one can either *enhance classification accuracy of neural networks to adversaries*, or devise *detectors to identify adversaries* in order to reject to process them. The former class of approaches, known as adversarial training, usually train a model on the training set, which is augmented by adversarial examples. The main difference between many adversarial training approaches lies in the way that the adversaries are created. For example, some [3,5,7,21] have trained the models with adversaries generated on-the-fly, while others conduct adversarial training with a pre-generated set of adversaries, either produced from an ensemble [22] or from a single model [18,23]. With the aim detecting adversaries to avoid making wrong decisions over the hostile samples, the second category of approaches propose the detectors, which are usually trained by a training set of adversaries [4,8–10,24,25].

Notwithstanding the achievement of some favorable results by both categories of approaches, the main concern is that their performances on all types of adversaries are extremely dependent on the capacity of generating an exhaustive set of adversaries, which comprises different types of adversaries. While making such a complete set of adversaries can be computationally expensive, it has been shown that adversely training a model on a specific type of adversaries does not necessarily confer a CNN robustness to other types of adversaries [11,12].

Some ensemble-based approaches [26,27] were shown to be effective for mitigating the risk of adversarial examples. Strauss et al. [26] demonstrated some ensembles of CNNs that are created by bagging and different random initializations are less fooled (misclassify adversaries), compared to a single model. Recently, Kariyappa et al. [27] have proposed an ensemble of CNNs, where they explicitly force each pair of CNNs to have dissimilar fooling directions, in order to promoting diversity in the presence of adversaries. However, computing similarity between the fooling directions by each pair of members for every given training sample is computationally expensive, results in increasing training time.

6 Conclusion

In this paper, we propose an ensemble of specialists, where each of the specialist classifiers is trained on a different subset of classes. We also devise a simple voting mechanism to efficiently merge the predictions of the ensemble's classifiers. Given the assumption that CNNs are strong classifiers and by leveraging diversity in this ensemble, a gap between predictive confidences of clean samples and those of black-box adversaries is created. Then, using a global fixed threshold, the adversaries predicted with low confidence are rejected (detected). We empirically demonstrate that our ensemble of specialists approach can detect a large portion of black-box adversaries as well as makes the generation of white-box attacks harder. This illustrates the beneficial role of diversity for the creation of ensembles in order to reduce the vulnerability to black-box and white-box adversarial examples.

Acknowledgements. This work was funded by NSERC-Canada, Mitacs, and Prompt-Québec. We thank Annette Schwerdtfeger for proofreading the paper.

References

1. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org (2017), pp. 1321–1330 (2017)
2. Eykholt, K., et al.: Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945 (2017)
3. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)

4. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. In: Proceedings of the 5th International Conference on Learning Representations (ICLR) (2017)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
6. Liao, F., Liang, M., Dong, Y., Pang, T., Zhu, J., Hu, X.: Defense against adversarial attacks using high-level representation guided denoiser. arXiv preprint arXiv:1712.02976 (2017)
7. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
8. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410 (2017)
9. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.: On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280 (2017)
10. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Advances in Neural Information Processing Systems, pp. 7167–7177 (2018)
11. Zhang, H., Chen, H., Song, Z., Boning, D., Dhillon, I.S., Hsieh, C.J.: The limitations of adversarial training and the blind-spot attack. arXiv preprint arXiv:1901.04684 (2019)
12. Tramèr, F., Boneh, D.: Adversarial training and robustness for multiple perturbations. In: Proceedings of the Neural Information Processing Systems (NeurIPS) (2019)
13. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770 (2016)
14. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
15. Charles, Z., Rosenberg, H., Papailiopoulos, D.: A geometric perspective on the transferability of adversarial directions. arXiv preprint arXiv:1811.03531 (2018)
16. Bendale, A., Boult, T.E.: Towards open set deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1563–1572 (2016)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
18. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
19. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE (2017)
20. He, W., Wei, J., Chen, X., Carlini, N., Song, D.: Adversarial example defenses: ensembles of weak defenses are not strong. arXiv preprint arXiv:1706.04701 (2017)
21. Huang, R., Xu, B., Schuurmans, D., Szepesvári, C.: Learning with a strong adversary. arXiv preprint arXiv:1511.03034 (2015)
22. Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., McDaniel, P.: Ensemble adversarial training: attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)
23. Rozsa, A., Rudd, E.M., Boult, T.E.: Adversarial diversity and hard positive generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 25–32 (2016)
24. Lu, J., Issaranon, T., Forsyth, D.: Safetynet: detecting and rejecting adversarial examples robustly. In: The IEEE International Conference on Computer Vision (ICCV), October 2017

25. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples (2017)
26. Strauss, T., Hanselmann, M., Junginger, A., Ulmer, H.: Ensemble methods as a defense to adversarial perturbations against deep neural networks. arXiv preprint arXiv:1709.03423 (2017)
27. Kariyappa, S., Qureshi, M.K.: Improving adversarial robustness of ensembles with diversity training. arXiv preprint arXiv:1901.09981 (2019)