LabelFool: A Trick In The Label Space

Yujia Liu
Department of Information and Computational Sciences
School of Mathematical Sciences, Peking University
Beijing, China
yujia_liu@pku.edu.cn

Ming Jiang
Department of Information and Computational Sciences
School of Mathematical Sciences, Peking University
Beijing, China
ming-jiang@pku.edu.cn

Tingting Jiang
National Engineering Research Center of Visual Technology
School of Computer Science, Peking University
Beijing, China
ttjiang@pku.edu.cn

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892136

***Abstract*—Adversarial attack methods can induce machine learning classifiers to mislabel errors. Current methods pay much attention to errors in the image space, i.e. the imperceptibility of adversarial perturbations, to avoid attacks being detected by humans. However, they overlook errors in the label space, i.e. the similarity between the wrong label and the true label. It is easy for humans to detect attacks if the wrong label has a big difference with the true label, for example, a dog is mislabeled as a cat. In this paper, we propose a novel attack method called LabelFool which attacks images with undetectable errors in both label space and image space. Given a classifier, for each input image, LabelFool first predicts the true label by estimating its probability distribution, then selects one label perceptually nearest to the predicted true label as the target label. Then LabelFool generates the adversarial sample by moving the input image towards the classification boundary between the predicted true label and the target label. The subjective experiments on ImageNet and visual results on CASIA-WebFace show that LabelFool is less detectable in the label space than other attack methods. Moreover, LabelFool has low perceptibility in the image space together with a high attack rate.**

***Index Terms*—adversarial attack, target label, imperceptibility**

I. Introduction

Deep neural networks are powerful learning models that achieve state-of-the-art pattern recognition performance in classification tasks [1], [2]. Nevertheless, it is found that adding well-designed perturbations to original samples can lead to mis-classification of deep neural networks [3]. These kinds of samples are called *adversarial samples*. The wrong labels of adversarial samples are called *target labels*. Techniques for generating adversarial samples are called *attack methods*.

To judge whether an attack method is good or not, the attack rate and the imperceptibility of attacks in the image space are two primary evaluation metrics [4]–[7]. The attack rate evaluates the effectiveness of an attack method. It calculates the ratio of input samples which are turned into adversarial samples by an attack method. To achieve a high attack rate, previous works have tried many optimization methods, such as gradient descent method [5], momentum method [8], Nesterov method [9] and so on. The imperceptibility of attacks in the image space affects how easily an attack method can be detected by humans. Low perceptibility means the perturbation added to the input sample is sufficiently small in the pixel level that people cannot notice. Previous works tried to minimize perturbations' L_0 norm [10], L_2 norm [6], [7] or L_∞ norm [5], [8], [9] to achieve low perceptibility in the image space.

However, it is not enough to consider the imperceptibility in the image space so that attacks cannot be detected by humans. The imperceptibility of attacks in the label space also affects how easily attacks are detected by humans. Here, the imperceptibility in the label space represents the "similarity" between the true label of an adversarial sample and its target label. For example (see Figure 1), if an Alaskan Malamute is mislabed as a cat, then the human detector will easily recognize the mistake and detect attacks. By contrast, if an Alaskan Malamute is mislabed as a Siberian Husky, then it will be harder for the human detector to find the error. Existing attack methods ignore the importance of target labels. Untargeted attacks generate adversarial samples without regarding to a target label [5], [6]. Targeted attacks only provide methods to generate adversarial samples with specified target labels, but they do not care how to select target labels. Usually, target labels in targeted attacks are chosen randomly [4], [7].

We want to go further on making the attack less detectable and focus on the imperceptibility in the label space. Label imperceptibility is important in some security tracks where a human is in the loop. For example, there is usually a human guard to check the results in facial recognition systems. In this paper, we propose an attack method called LabelFool to make attacks hard to be detected in both image space and label space. It contains two parts as Figure 2 shows. The first part is "label selection" which is in charge of choosing target labels for input samples. Specifically, the label selection algorithm is designed to choose target labels which are perceptually similar to the ground truth labels because they are less detectable in the label space. If the ground truth label is not known, LabelFool will speculate the true label through a probability model first. The second part is "sample generation" which means generating the adversarial samples so that they are mis-classified as the target

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

labels. As for the sample generation part, LabelFool refers to the thought in DeepFool [6], that is, moving the input towards the classification boundary between the current class and the target class. To evaluate the imperceptibility of attacks in the label space, we also propose two metrics – confusion rate and real confusion rate (introduced in Section IV-E).

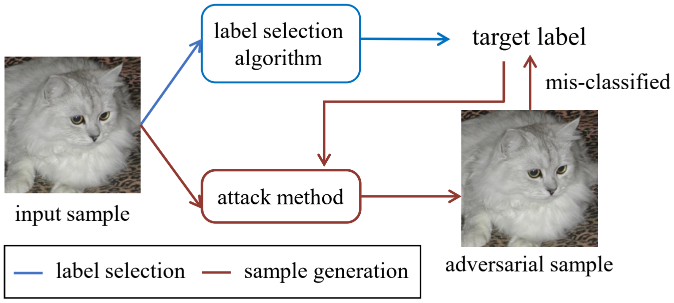


Fig. 2. The outline of LabelFool. LabelFool contains two parts. The first part is "label selection" where a less detectable target label is chosen by an algorithm. The second part is "sample generation" where attacks add perturbations to the input sample so that it can be mis-classified as the specific label.

To demonstrate the good performance of LabelFool confusing people in the label space, we conduct subjective experiments on ImageNet [11] and show some visual results on CASIA-WebFace [12]. For the integrity of the experiment, we also report LabelFool's performance on traditional metrics– the attack rate and the imperceptibility in the image space. Compared with FGSM [5], DeepFool and SparseFool [13], LabelFool is about 1-8% higher than other methods on the confusion rate and real confusion rate. At the same time, LabelFool guarantees low imperceptibility in the image space and maintains a high attack rate.

II. Related Work

The phenomenon that neural networks are sensitive to adversarial samples was observed by [3]. Since then, many researchers study how to generate adversarial samples. Meanwhile, making the perturbations in the image space as imperceptible as possible becomes a default requirement for attackers. However, few works focused on the imperceptibility of adversarial perturbations in the label space.

A. Untargeted Attack

Untargeted attack methods are proposed to meet the need that attackers just want to generate samples mis-classified by networks without any other requirements. FGSM [5] uses gradient sign method to generate adversarial samples. One-pixel attack [10] and SparseFool [13] are two methods attacking networks in a scenario where attackers perturb one pixel/a few pixels and lead to mis-classifications.

These works do not care about the target labels at all. This will cause an apparent mis-classification so that humans will sound the defensive alarm quickly. DeepFool [6] generates adversarial samples by directly moving the input sample to the nearest class in the feature space. The mis-classified labels for DeepFool are related to the ground truth labels to some extent, because features extracted from classification models can reflect images' perceptual information. Moreover, the classes which are close in the feature space are often perceptually similar. However, DeepFool approximates the multi-dimensional classification boundaries in two dimensions and this might make big errors on finding the nearest class.

B. Targeted Attack

Targeted attacks aim to generate adversarial samples which are mis-classified as specific target labels, but they do not tell how to chose target labels. There is no way to annotate target labels for each input sample in practical applications because of the annotation cost. CW attack [4] is the first proposed method that can cause targeted mis-classification on the ImageNet dataset. This method generates adversarial samples by solving an optimization problem based on L_p constraint, and it chooses target labels through three simple methods, i.e. choosing a

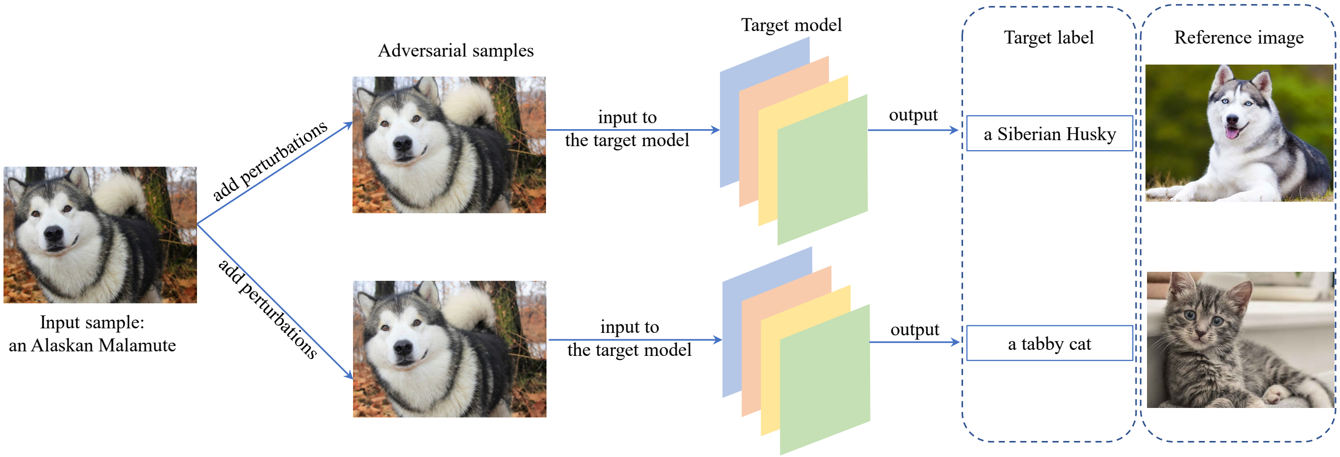


Fig. 1. The input image is an Alaskan malamute, the perceptibility of attacks in the label space is low if the target label of the adversarial sample is "Siberian Husky" because these two kinds of dogs look similar. Otherwise, attacks will be easily detected if the target label is "tabby cat". In the last column of the figure, we show the reference image of the corresponding target label.

random label or choosing the least/most difficult label. TR attack [7] uses the trust region algorithm to realize the targeted attack and utilizes the same ways as CW attack to choose target labels. However, these simple ways blindly choose target labels and the authors did not give any scenes where these label chosen methods are needed.

In this paper, we focus on the importance of adversarial samples' target labels and fill the gap that no works give a guide about how to choose target labels appropriately. We propose a novel attack method called LabelFool to choose less detectable target labels and generate adversarial samples efficiently. We compare LabelFool with FGSM, DeepFool and SparseFool, to show that LabelFool is helpful in making the attack less detectable by humans in the label space. We also demonstrate that the performance gain in the label space is not at the expense of the loss in the image space or attack rate.

III. LabelFool

In this section, we will introduce the label selection part (Section III-A) and the sample generation part (Section III-B) in LabelFool respectively. We use the same notation i ($i = 1, 2, \dots$) for "class" and "label", because "class" and "label" are interchangeable in this paper.

A. Label Selection

Figure 3 shows the pipeline of label selection part for LabelFool, whose task is choosing the target label t for an input image x. LabelFool aims to find the most nearest label to the input image's ground truth. There are two problems to be solved: (1) how to define the distance between labels? (2) given the ground truth label g is unknown, how to choose the nearest label to g? In this subsection, we will solve these problems one by one.

For the first problem, we use the perceptual distance to measure the distance between images, and utilize Hausdorff distance to calculate the distance between labels. In detail, given two images x, y, we first use pre-trained image classification models extract perceptual features ϕ_x, ϕ_y for x, y. Because features extracted by pre-trained image classification models can reflect some perceptual information, Then we follow previous works [14], [15] which use cosine distance to measure perceptual distance between x and y, i.e. $d(x, y) = 1 - \cos(\phi_x, \phi_y)$.

After calculating the distance between images, we can compute the distance between labels. Each label corresponds to a set of images. To measure the distance between two sets, Hausdorff distance [16] is a good choice. The distance between label i and label j is denoted as $\boldsymbol{D}_{i,j}$. For a dataset with n labels, a matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ can be constructed by calculating the distance between all pairs of labels in the dataset. Remark that the computation of $\boldsymbol{D}$ considers all training images in a dataset and $\boldsymbol{D}$ is stored for further use. Then, the first problem, how to define the distance between labels, is solved.

There are two steps to solve the second problem, i.e. given the ground truth label g is unknown, how to choose the nearest label to g? First, we estimate the probability distribution of g by the output of the target model (i.e. the model to be attacked) and denote the estimated distribution as $\hat{P}$. According to $\hat{P}$, there are two cases. For the first case where the maximum value in $\hat{P}$ is large, we specify the label with the highest probability as the ground truth. Then, it is easy to get the nearest label according to the matrix $\boldsymbol{D}$. For the second case where the maximum value in $\hat{P}$ is small, we estimate the expected distance between label i and g under the distribution $\hat{P}$. Then the label with the minimum expected distance is chosen as the target label t.

In detail, let x be an image whose ground truth is g and f be a target model. The output of f is a probability vector $f(x) = (\hat{p}_1, \dots, \hat{p}_n)^T$ where $\hat{p}_i$ represents the probability of x belonging to class i, i.e. $P(x \in class\ i) = \hat{p}_i$. Therefore, $f(x)$ can be taken as the estimated probability distribution of g, i.e. $\hat{P} = f(x)$. For simplicity, suppose $\hat{p}_1, \dots, \hat{p}_n$ are sorted in the descending order.

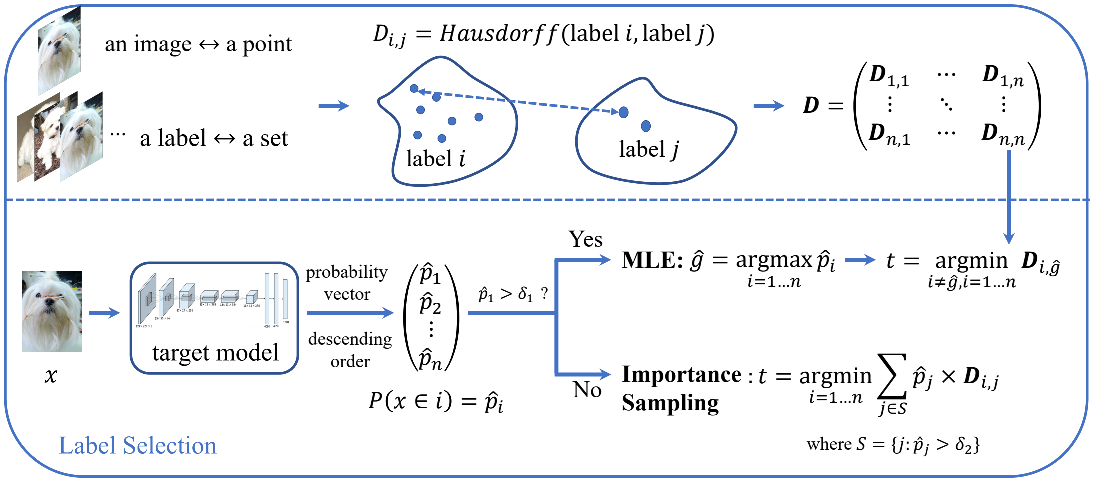

an image ↔ a point
··· a label ↔ a set
$D_{i,j} = Hausdorff(\text{label } i, \text{label } j)$
label i
label j
$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_{1,1} & \cdots & \boldsymbol{D}_{1,n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{D}_{n,1} & \cdots & \boldsymbol{D}_{n,n} \end{pmatrix}$
x
target model
probability vector
descending order
$\begin{pmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \vdots \\ \hat{p}_n \end{pmatrix}$
$P(x \in i) = \hat{p}_i$
$\hat{p}_1 > \delta_1$?
Yes
No
MLE: $\hat{g} = \underset{i=1\dots n}{\text{argmax}}\, \hat{p}_i$
$t = \underset{i \neq \hat{g}, i=1\dots n}{\text{argmin}}\ \boldsymbol{D}_{i,\hat{g}}$
Importance Sampling : $t = \underset{i=1\dots n}{\text{argmin}} \sum_{j \in S} \hat{p}_j \times \boldsymbol{D}_{i,j}$
where $S = \{j : \hat{p}_j > \delta_2\}$
Label Selection

Fig. 3. **Label selection algorithm for LabelFool**, we first compute the distance $\boldsymbol{D}_{i,j}$ between every two class i, j in a n-classes dataset. Then we choose the target label t_x for an input image x by two strategies according to the value of $\hat{p}_1$.

(1) When $\hat{p}_1$ is larger than some threshold δ_1, we use Maximum Likelihood Estimation (MLE) [17] to estimate g. That is, we take

$$\hat{g} = \arg\max_{i=1,\ldots,n} \hat{p}_i \tag{1}$$

as the ground truth g. Then, the target label t is chosen as

$$t = \arg\min_{i \neq \hat{g}, i=1,\ldots,n} \boldsymbol{D}_{i,\hat{g}} \quad \text{if } \hat{p}_1 > \delta_1. \tag{2}$$

(2) When $\hat{p}_1$ is smaller than the threshold δ_1, the target model f is not sure which label is true and meanwhile, it is hard to sample from the real probability distribution. According to Importance Sampling [18], some important labels can be sampled to estimate the expected distance between label i and the ground truth g under $\hat{P}$. Specifically, the sampled labels are

$$S = \{j : \hat{p}_j > \delta_2\}, \tag{3}$$

i.e., labels with probability larger than some threshold δ_2. Then, the distance between the label i and the ground truth g (with $g \sim \hat{P}$) can be estimated by

$$\sum_{j \in S} \hat{p}_j \cdot \boldsymbol{D}_{i,j}. \tag{4}$$

Therefore, the target label is chosen as

$$t = \arg\min_{i=1,\ldots,n} \sum_{j \in S} \hat{p}_j \cdot \boldsymbol{D}_{i,j} \quad \text{if } \hat{p}_1 \leq \delta_1. \tag{5}$$

In conclusion, the whole strategy for choosing the target label t of an input image x is computed as

$$t = \begin{cases} \arg\min\limits_{i \neq \hat{g}, i=1,\ldots,n} D_{i,\hat{g}} & \text{if } \hat{p}_1 > \delta_1 \\ \arg\min\limits_{i=1,\ldots,n} \sum_{j \in S} \hat{p}_j \cdot D_{i,j} & \text{otherwise} \end{cases} \tag{6}$$

where S is formulated in Eq. (3).

B. Sample Generation

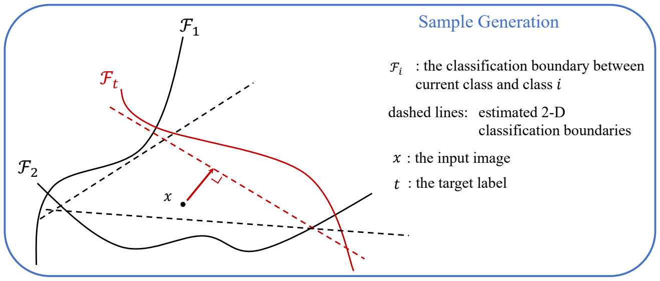
Figure 4 shows the pipeline of the sample generation part for LabelFool. To achieve a relatively small perturbation in the image space and a relatively fast speed to generate adversarial samples, we design a method based on DeepFool. The mathematical derivation in this step is similar to DeepFool [6] and the only difference is that, we have a target label while DeepFool does not.

As introduced in DeepFool [6], a high dimensional classification boundary can be approximated by a line in two dimensions. Given an image x_0 and a target model f, the 2D approximated boundary classification boundary between the current class $\hat{g}$ and the target class t is

$$\mathcal{F}_t = \{x : \nabla f_{\hat{g}}(x_0)x - \nabla f_t(x_0)x + f_{\hat{g}}(x_0) - f_t(x_0) = 0\}. \tag{7}$$

Therefore, to make x_0 misclassified as t, the perturbation with minimum L_2 norm is: moving x_0 towards the direction

$$\frac{\nabla f_{\hat{g}}(x_0) - \nabla f_t(x_0)}{\|\nabla f_{\hat{g}}(x_0) - \nabla f_t(x_0)\|_2^2}, \tag{8}$$



Sample Generation
$\mathcal{F}_1$
$\mathcal{F}_t$
$\mathcal{F}_2$
x
$\mathcal{F}_i$: the classification boundary between current class and class i
dashed lines: estimated 2-D classification boundaries
x : the input image
t : the target label

Fig. 4. **Sample generation step for LabelFool**. LabelFool moves the input towards the boundary $\mathcal{F}_t$, which is the boundary between the current class and the target class. This operation will be repeated several times until it is classified as t.

and the distance traveled is

$$|f_{\hat{g}}(x_0) - f_t(x_0)|. \tag{9}$$

We repeatedly move the current point towards $\mathcal{F}_t$ until it is classified as label t or the maximum number of iterations has been reached. A pseudo-code of the sample generation part is shown in Algorithm 1.

```
Algorithm 1 : Sample Generation
Input: image $x$, target model $f$, target label $t$
Output: Adversarial image $\hat{x}$
initialize $x_0 \leftarrow x, i \leftarrow 0, f(x_0) = (\hat{p}_1, \ldots, \hat{p}_n)^T$
initialize $\hat{g} \leftarrow \arg\max_i \hat{p}_i$
while $\hat{g} \neq t$ and $i < max\_iter$ do
    $w \leftarrow \nabla f_{\hat{g}}(x_i) - \nabla f_t(x_i)$
    $k \leftarrow f_{\hat{g}}(x_i) - f_t(x_i)$
    $r_i \leftarrow \frac{|k|}{\|w\|_2^2} w$
    $x_{i+1} \leftarrow x_i + r_i$
    $f(x_{i+1}) = (\hat{p}_1, \ldots, \hat{p}_n)^T$
    $\hat{g} \leftarrow \arg\max_i \hat{p}_i, i \leftarrow i + 1$
end while
Return $\hat{x} = x_{i+1}$
```

IV. Experiments

We evaluate our method on two datasets: ImageNet [11] and CASIA-WebFace [12].

ImageNet provides the CLS-LOC dataset for classification tasks. Its train split contains about 1,300,000 images for 1,000 classes. Every class has a label, it is a word from WordNet. Our experiments are conducted on the train split of the dataset.

CASIA-WebFace has nearly 500,000 face images for 10,575 people. Every person has an ID, which is a number from 1 to 10,575 (not a name), and several reference images. So for this dataset, target label represents the target ID. This dataset is chosen to show that LabelFool is effective on face datasets. As the technique of face recognition is widely used in our daily life, undetectable (in both image space and label space) attacks will cause potential security problems in real life. Therefore, attacks with high imperceptibility in the label space are stronger than those with low imperceptibility.

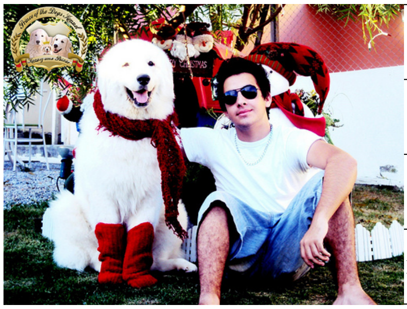
We first propose two metrics to measure the imperceptibility of attacks in the label space and perform extensive experiments to show LabelFool is less detectable than other attacks in the label space (Section IV-A). Meanwhile, LabelFool has good performance in the image space (Section IV-B). In Section IV-C, we conduct attacks on several models to prove the exceptional ability of LabelFool on attacking neural networks. Computational cost is analyzed in Section IV-D. Section IV-E introduces the ablation study where we compare the label selection method in LabelFool and the simple selection methods proposed by CW attack [4].

A. *Imperceptibility in the Label Space*

In this part, we will show the effectiveness of LabelFool on being less detectable by human observers in the label space. We will compare LabelFool with three attack methods: DeepFool [6], FGSM [5] and SparseFool [13].

Newly proposed metrics. As no works have measured the imperceptibility of attacks in the label space, we propose two metrics, Confusion Rate (CR) and Real Confusion Rate (RCR), for the imperceptibility in the label space. These metrics are defined by how well human observers solve the *puzzles* in subjective experiments. A puzzle is the combination of an successfully-attacked adversarial image and its label. In the subjective experiment, a human observer needs to determine whether the label is correct for the image, answering “True” or “False” for each puzzle. The rate with which the observer answers incorrectly is called *Confusion Rate (CR)*.

If the object corresponding to the target label in a wrong-answered puzzle indeed exists in the image, we call it a *fake attack*. Figure 5 illustrates what is a fake attack. In Figure 5, the main object of the image is a dog but fake attacks would choose “sunglasses” as the target label. However, there is a man wearing sunglasses in the image, so “sunglasses” is not exactly a wrong label.



	Label
Ground Truth	Kuvasz
Fake Attack	Sunglasses
LabelFool	Great Pyrenees

Fig. 5. An example of fake attacks. The main object of the image is a dog whose ground truth is “Kuvasz”. Fake attacks choose “sunglasses” as the target label. However, sunglasses indeed exist in the image. By contrast, LabelFool aims to choose a target label which is perceptually similar to the ground truth label. In this example, LabelFool choose “Great Pyrenees” as the target label.

We annotate every wrong-answered puzzle whether it is a fake attack. After getting rid of all fake attacks, the observer has a new confusion rate, we call it *Real Confusion Rate(RCR)*. Figure 6 shows how to compute CR and RCR intuitively. The higher of these two evaluations, the less detectable of attacks in the label space.

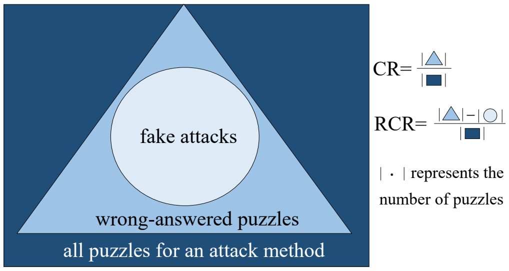

fake attacks
wrong-answered puzzles
all puzzles for an attack method
CR= |▲| / |■|
RCR= |▲|−|○| / |■|
| · | represents the number of puzzles

Fig. 6. A diagram about how to compute confusion rate and real confusion rate. Here |·| means “the number of”.

Experiments on ImageNet. We sample 600 source images from ImageNet split randomly which belong to different classes. Each source image will derive four adversarial images by attacking ResNet-50 [19], namely DeepFool-attacked image, LabelFool-attacked image, FGSM-attacked image and SparseFool-attacked image. As the term “puzzles” is used to describe the combination of an image and its label, this experiment has $4 \times 600 = 2400$ puzzles. The evaluations are CR and RCR and ther are 10 observers (3 females and 7 males, age between 20-29) doing the subjective experiment. Average CR and RCR are reported in Figure 7.

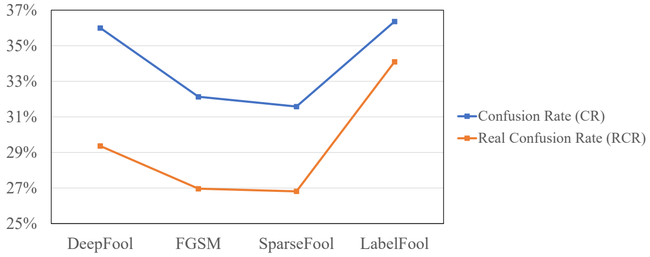


Fig. 7. A line chart for average confusion rate (CR) and real confusion rate (RCR) of 10 observers. The horizontal axis represents four attack methods. The vertical axis represents the mean value of 10 human observers' evaluations.

From Figure 7, we can see LabelFool wins in both evaluations among all attack methods. Especially, there is a huge improvement in RCR compared with other attack methods, about 5-8 percent improvement. DeepFool achieves a comparable CR with LabelFool but the RCR of DeepFool is poor[1]. The drop for DeepFool from a relatively high CR to a low RCR reflects that DeepFool has many fake attacks. Compared with FGSM and SparseFool, LabelFool is ahead in both evaluation.

Experiments on CASIA-WebFace. In this experiment, we choose 1,000 images from CASIA-WebFace randomly which have different IDs. Every sampled image will be fed into

[1] The RCR of DeepFool is significant lower than LabelFool by the hypothesis significance test (p-value is 0.0055).

SphereFace [20] model and then, we use LabelFool, DeepFool, FGSM and SparseFool to attack the model respectively. Each attack method can generate one adversarial sample with a target ID. Therefore, 4 target IDs can be got for one input image. Then one reference image is chosen for each target ID from its reference images in the dataset. To evaluate the effectiveness of attack methods in the label space, it is needed to judge whether the person in the target ID's reference image is the same person in the input image.

In terms of evaluation, we do not conduct subject experiments to compute CR/RCR for two reasons. First, it is hard for normal people to judge whether the people in two images are the same. Second, we have conducted some preliminary experiments which show that there is huge individual difference between human observers for this task. Instead, to show LabelFool's imperceptibility in the label space, we show two visual results in Figure 8. The first column of Figure 8 is the input image and its ID. The second column to the last column are the ID generated by DeepFool, LabelFool, FGSM, SparseFool respectively and the ID's reference picture.

In Figure 8, we can see that the person with the ID chosen by LabelFool looks most like the one with the ground truth ID among all people with target IDs. Some methods even get an ID whose sex is different from the real ID, such as FGSM in the second visual example. We show more visual examples on CASIA-WebFace in our supplementary materials to demonstrate the excellence of LabelFool in being less detectable in the label space.

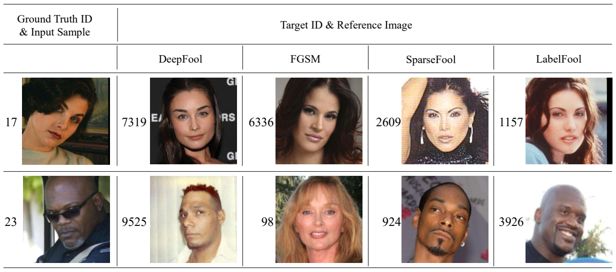


Fig. 8. Visual results on CASIA-WebFace. The first column shows the input image and its ground truth ID. The second to the last columns show the target ID (a number) generated by DeepFool, LabelFool, FGSM and SparseFool respectively. The image beside the ID is a reference image for this ID.

B. Imperceptibility in the Image Space

In this subsection, we will show our performance in the image space to demonstrate that our improvement in the label space is not at the cost of huge loss in the image space. We use the L_2 norm of perturbations to evaluate the imperceptibility in the image space. The definition is $\|\Delta\| := \frac{1}{W_N \times H_N}\sum_{w=1}^{W_N}\sum_{h=1}^{H_N}\|\Delta y_{w,h}\|^2$, where $W_N \times H_N$ is the size of the input image, $\Delta y_{w,h}$ represents the perturbation added to the pixel $y_{w,h}$. This evaluation is also utilized in previous works [3], [6]. The smaller $\|\Delta\|$ is, the better the adversarial samples are.

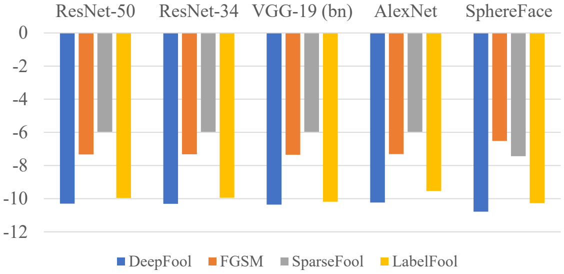


Fig. 9. Log of the mean value of perceptibility for adversarial samples generated by different attack methods on different models. The x-axis represents target model and the y-axis represents $\ln(\text{mean}(\|\Delta\|))$. Different colors represent different attack methods. The smaller $\ln(\text{mean}(\|\Delta\|))$ represents the better imperceptibility.

We randomly choose 1000 images from ImageNet and 1000 images from CASIA-WebFace, and then attack the classifier to generate adversarial samples. We compute mean value of $\|\Delta\|$ for these adversarial samples. In this experiment, we test four classifiers for ImageNet: ResNet-34 [19], ResNet-50 [19], VGG-19 (with batch normalization) [21] and AlexNet [1], and one classifier for CASIA-WebFace: SphereFace [20]. The results are shown in Figure 9. We can see although LabelFool is significantly better than FGSM and SparseFool. Although LabelFool is a little worse than DeepFool, visual results (Figure 10) indicate that human observers can not notice the difference between LabelFool and DeepFool in the image space.

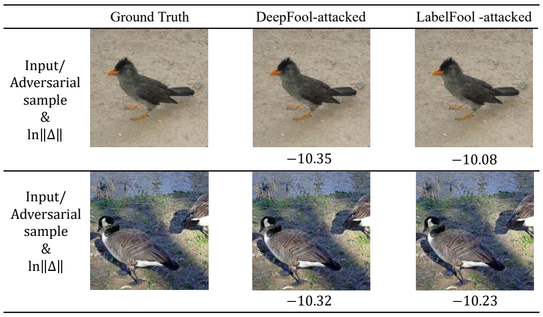


Fig. 10. Two visual results of adversarial samples generated by DeepFool and LabelFool against AlexNet with $\ln\|\Delta\|$ reported below. Although $\ln\|\Delta\|$ of LabelFool is a little higher, we can hardly distinguish the difference between adversarial samples generated by LabelFool and DeepFool.

C. Attack Rate

We will show the attack rate in the last experiment which is a fundamental requirement for an attack method. Results are shown in Table I. As for ImageNet, we randomly choose 3000 original images and use these original images to generate adversarial images for ResNet-34, ResNet-50, VGG-19 (with batch normalization) and AlexNet respectively. As for CASIA-WebFace, 3000 original images are chosen and then 3000 adversarial images are generated for SphereFace. We surprisingly find that LabelFool has the highest attack rate on all models compared with other methods. This might benefit from our probability model which is used to choose the target label. Because in our strategy, when $\hat{p}_1 \leq \delta_1$, we do not use the predicted label as the ground truth. Instead, we consider all labels whose probability are larger than δ_2 and choose the label nearest to all these labels as the target label. This operation can avoid some mistakes and improve the attack rate when the classifier doesn't give a correct classification result.

TABLE I
ATTACK RATE OF DIFFERENT METHODS ON DIFFERENT MODELS.

Model	**DeepFool**	**FGSM**	**SparseFool**	**LabelFool**
Dataset: ImageNet				
ResNet-34	92.7%	95.0%	92.6%	**97.5%**
ResNet-50	93.1%	95.1%	92.5%	**97.9%**
VGG-19(bn)	92.0%	94.6%	83.7%	**97.5%**
AlexNet	90.4%	96.4%	89.1%	**97.4%**
Dataset: CASIA-WebFace				
SphereFace	98.7%	99.2%	97.8%	**99.3%**

D. Computational Cost

The computational cost of LabelFool consists of two parts. The first part is the cost of an extra preprocessing step for images to compute the matrix $\boldsymbol{D}$ in the label selection step. The computational complexity of this process is $\mathcal{O}(n^2)$ where n is the number of images in the training set. Although computing $\boldsymbol{D}$ takes some time, it is only computed once for each dataset and stored for further uses.

The second part is the cost of generating adversarial examples in the sample generation step. This process is common to all attack methods. We calculate the average time for different methods to generate an adversarial example over 3000 ImageNet images against ResNet50 in Table II. Our experiment was conducted with Intel(R) Core(TM) i7-8700K CPU @ 3.70Hz CPU and NVIDIA GeForce GTX 1080. The operating system is Ubuntu 16.04.7 LTS. From Table II, it can be seen that LabelFool has comparable computation time to DeepFool.

TABLE II
COMPUTATION TIME OF DIFFERENT METHODS WHERE "S" REPRESENTS SECOND.

Method	**DeepFool**	**FGSM**	**SparseFool**	**LabelFool**
Time per image	0.91s	0.08s	9.54s	0.33s

E. Ablation Studies

In this subsection, we compare our label selection method (Section 3) with two common simple label selection methods proposed in CW attack. They are:

- *Random label*: select the target label uniformly at random among the labels that are not the correct label.
- *Easiest label*: select the target label that is least difficult to attack. That is to say, choose the second highest label of the output $f(x)$.

In particular, we replace our label selection methods with these methods and attack in the same way to generate adversarial samples We design a toy subjective experiment to show the drawbacks of these simple methods. We first sample 600 source images from ImageNet randomly, and these images are in different classes. The target model is ResNet-50 [19]. We use the easiest label, random label and our label selection method respectively in the label selection part, and use the same method (method in Algorithm 1) in the sample generation part. Each source image will derive three adversarial images, and each adversarial image has its target label. There are $3 \times 600 = 1800$ puzzles in this experiment.

From Table III, the drawbacks for simple methods are obvious. Random label has an extremely low CR which means it can hardly confuse people. We did not annotate fake attacks for random labels' puzzles because CR is a upper bound of RCR, and CR for random labels' puzzles is too low to compare with other methods. Easiest label method is less detectable than random label method, but it has a poor performance on RCR which means the objects corresponding to the easiest labels usually indeed exist in the image[2]. That is, there are many fake attacks in easiest labels' puzzles. This feature of easiest label may be useful in some special applications, but in this paper, we need a high RCR on which easiest label is not as good as our label selection method.

TABLE III
CR AND RCR FOR DIFFERENT LABEL SELECTION METHODS.

Method	**Confusion Rate**	**Real Confusion Rate**
Random	0.83%	-
Easiest	**49.17%**	38.00%
Ours	44.00%	**41.83%**

V. CONCLUSION AND FURTHER DISCUSSION

Conclusion. In this study, we observe two important issues: (1) the target label selection is necessary for attackers and (2) how to choose a less detectable target label.

As for the first issue, the target label selection is important because when attacks are utilized in real life, attackers usually have some special needs on target labels. While previous works do not care about the target label selection, we add the target label selection as a part of our attack method. Actually, the target label selection part in this paper is a flexible module, because attackers can choose or design label selection methods according to their actual needs.

For the second issue, a natural need for attackers is making attacks undetectable by humans. Therefore, we provide

[2] The RCR of our method is significantly higher than the easiest label selection method by the hypothesis significance test (p-value is 0.0274).

LabelFool to identify a target label perceptually similar to an input image's ground truth, so that a human observer will overlook the mis-classification. Our experiments show that, LabelFool's *CR* and *RCR* is 2-8 percentage higher than other methods on ImageNet. Meanwhile, the perceptibility of LabelFool in the image space, measured by the mean L_2 norm of perturbations, is low. As for attack rate, LabelFool is the highest among compared methods.

Further discussion. In this paper, we just consider the need for attacks being undetectable by humans and propose a feasible way to generate adversarial samples which can confuse people in the label space. However, there are other needs in some important and special applications. For example, to ensure the security of network content, we usually use neural networks to check whether a message contains violence, eroticism or other undesirable content. In this applications, an attacker may hope to attack undesirable messages to be mis-classified as healthy messages and do great harm to network security. Works can be carried out on such needs.

For LabelFool, there is still some room for improvement. First, the computational complexity of this process is $\mathcal{O}(n^2)$ where n is the number of images in the training set. Although $\boldsymbol{D}$ is computed once for each dataset and it is stored for further uses, the computational time is still high which needs to be optimized. Second, we only consider perceptual distance in this paper, but semantic distance also has its significance of confusing people in the label space. We may take the semantic tree into consideration and make a trade off between perceptual distance and semantic distance in future research.

Our results provide the following avenues for future research.

- Adversarial attacks are real threat to both human users and networks designers when they can be used into real applications. Therefore, considering how to make attacks applicable may be more urgent than how to generate an adversarial sample.
- The label selection part is more important for attacks than the sample generation part as there have been many optional methods for the sample generation part. If we can categorize the needs in real applications and design general label selection methods for each category, it will be very convenient to make attacks applicable in applications.

Acknowledgements

This work was partially supported by Sino-German Center (M 0187) and Natural Science Foundation of China (11961141007 and 62088102). We also acknowledge High-Performance Computing Platform of Peking University for providing computational resources.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[2] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2010, pp. 253–256.

[3] 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.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE Symposium on Security and Privacy*, 2016, pp. 39–57.

[5] I. Goodfellow, S. Jonathon, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference of Learning Representation*, 2015.

[6] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[7] Z. Yao, A. Gholami, P. Xu, K. Keutzer, and M. W. Mahoney, "Trust region based adversarial attack on neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11350–11359.

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

[9] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," in *Proceedings of the International Conference on Learning Representations*, 2020.

[10] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[11] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[12] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[13] A. Modas, S. M. Moosavi-Dezfooli, and P. Frossard, "SparseFool: A few pixels make a big difference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9087–9096.

[14] T. Y. Lin, A. Roychowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2016, pp. 1449–1457.

[15] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5022–5030.

[16] J. Henrikson, "Completeness and total boundedness of the hausdorff metric," *MIT Undergraduate Journal of Mathematics*, vol. 1, pp. 69–80, 1999.

[17] M. L. Eaton, "Parametric statistical theory (johann pfanzagl)," *SIAM Review*, vol. 38, no. 3, pp. 527–529, 1996.

[18] A. Owen and Y. Zhou, "Safe and effective importance sampling," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 135–143, 2000.

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

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

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