Generalizable Data-free Objective for Crafting Universal Adversarial Perturbations

Konda Reddy Mopuri*, Aditya Ganeshan*, R. Venkatesh Babu, *Senior Member, IEEE*

Abstract—Machine learning models are susceptible to adversarial perturbations: small changes to input that can cause large changes in output. It is also demonstrated that there exist input-agnostic perturbations, called universal adversarial perturbations, which can change the inference of target model on most of the data samples. However, existing methods to craft universal perturbations are (i) task specific, (ii) require samples from the training data distribution, and (iii) perform complex optimizations. Additionally, because of the data dependence, fooling ability of the crafted perturbations is proportional to the available training data. In this paper, we present a novel, generalizable and data-free approaches for crafting universal adversarial perturbations. Independent of the underlying task, our objective achieves fooling via corrupting the extracted features at multiple layers. Therefore, the proposed objective is generalizable to craft image-agnostic perturbations across multiple vision tasks such as object recognition, semantic segmentation, and depth estimation. In the practical setting of black-box attack scenario (when the attacker does not have access to the target model and it's training data), we show that our objective outperforms the data dependent objectives to fool the learned models. Further, via exploiting simple priors related to the data distribution, our objective remarkably boosts the fooling ability of the crafted perturbations. Significant fooling rates achieved by our objective emphasize that the current deep learning models are now at an increased risk, since our objective generalizes across multiple tasks without the requirement of training data for crafting the perturbations. To encourage reproducible research, we have released the codes for our proposed algorithm[†].

Index Terms—Adversarial perturbations, fooling CNNs, stability of Neural Networks, perturbations, universal, generalizable attacks, attacks on ML systems, data-free objectives, adversarial noise.

1 Introduction

Small but structured perturbations to the input, called adversarial perturbations, are shown ([1], [2], [3]) to significantly affect the output of machine learning systems. Neural network based models, despite their excellent performance, are observed ([4], [5], [6]) to be vulnerable to adversarial attacks. Particularly, Deep Convolutional Neural Networks (CNN) based vision models ([7], [8], [9], [10], [11]) can be fooled by carefully crafted quasi-imperceptible perturbations. Multiple hypotheses attempt to explain the existence of adversarial samples, viz. linearity of the models [5], finite training data [12], etc. More importantly, the adversarial perturbations generalize across multiple models. That is, the perturbations crafted for one model fools another model even if the second model has a different architecture or is trained on a different dataset ([4], [5]). This property of adversarial perturbations enables potential intruders to launch attacks without the knowledge about the target model under attack: an attack typically known as *black-box attack* [13]. In contrast, an attack where everything about the target model is known to the attacker is called a *white-box attack*. Until recently, all the existing works assumed a threat model in which the adversaries can directly feed input to the machine learning system. However, Kurakin *et al.* [14] lately showed that the adversarial samples can remain misclassified even if they were constructed in physical world and observed through a sensor (e.g., camera). Given that the models are vulnerable even outside the laboratory setup [14], the models' susceptibility poses serious threat to their deploy-ability in the real world (e.g., safety concerns for autonomous driving). Particularly, in case of critical applications that involve safety and security, reliable models need to be deployed to stand against the strong adversarial attacks. Thus, the effect of these structured perturbations has to be studied thoroughly in order to develop dependable machine learning systems.

Recent work by Moosavi-Dezfooli *et al.* [8] presented the existence of image-agnostic perturbations, called universal adversarial perturbations (UAP) that can fool the state-of-the-art recognition models on most natural images. Their method for crafting the UAPs, based on the DeepFool [7] attacking method, involves solving a complex optimization problem (eqn. 2) to design a perturbation. The UAP [8] procedure utilizes a set of training images to iteratively update the universal perturbation with an objective of changing the predicted label upon addition. Similar to [8], Metzen *et al.* [11] proposed UAP for semantic segmentation task. They extended the iterative FGSM [5] attack by Kurakin *et al.* [14] to change the label predicted at each pixel. Additionally, they craft image-agnostic perturbations to fool the system in order to predict a pre-determined target segmentation output.

However, these approaches to craft UAPs ([8], [11], [15]) have the following important drawbacks:

- *Data dependency*: It is observed that the objective presented by [8] to craft UAP requires a minimum

- *The authors are with the Video Analytics Lab, Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India, 560012.*
E-mail: kondamopuri@iisc.ac.in, adityaganeshan@gmail.com and venky@iisc.ac.in

* denotes equal contribution

[†] https://github.com/val-iisc/gd-uap

number of training samples for it to converge and craft an image-agnostic perturbation. Moreover, the fooling performance of the resulting perturbation is proportional to the available training data (Figure 8). Similarly, the objective for semantic segmentation models (e.g., [11]) also requires data. Therefore, existing procedures can not craft perturbations when enough data is not provided.

- *Weaker black-box performance*: Since information about the target models is generally not available for attackers, it is practical to study the black-box attacks. Also, black-box attacks reveal the true susceptibility of the models, while white-box attacks provide an upper bound on the achievable fooling. However, the black-box attack of UAP [8] is significantly weaker than their white-box attack (Table 8). Note that, in [8], authors have not analyzed the performance of their perturbations in the black-box attack scenario. They have assumed that the training data of the target models is known and have not considered the case in which adversary has access to only a different set of data. This amounts to performing only *semi white-box* attacks. Black-box attacks generally imply ([13]) that the adversary does not have access to (i) the target network architecture (including the parameters), and (ii) a large training dataset. Even in the case of semantic segmentation, since [11] work with targeted attacks, they observed that the perturbations do not generalize to other models very well.
- *Task specificity*: The current objectives to craft UAPs are task specific. The objectives are typically designed to suit the underlying task at hand since the concept of fooling varies across the tasks. Particularly, for regression tasks such as depth estimation and crowd counting, extending the existing approaches to craft UAPs is non-trivial.

In order to address the above shortcomings and to better analyze the stability of the models, we present a novel data-free objective to craft universal adversarial perturbations, called *GD-UAP*. Our objective is to craft image-agnostic perturbations that can fool the target model without any knowledge about the data distribution, such as, the number of categories, type of data (e.g., faces, objects, scenes, etc.) or the data samples themselves. Since we do not want to utilize any data samples, instead of an objective that reduces the confidence to the predicted label or flip the predicted label (as in [4], [7], [8], [11]), we propose an objective to learn perturbations that can adulterate the features extracted by the models. Our proposed objective attempts to over-fire the neurons at multiple layers in order to deteriorate the extracted features. During the inference time, the added perturbation misfires the neuron activations in order to contaminate the representations and eventually lead to wrong prediction.

This work extends our earlier conference paper [9]. We make the following new contributions in this paper:

1) We propose a novel data-free objective for crafting image-agnostic perturbations.
2) We demonstrate that our objective is generalizable across multiple vision tasks by extensive evaluation of the crafted perturbations across three different vision tasks covering both classification and regression.
3) Further, we show that apart from being data-free objective, the proposed method can exploit minimal prior information about the training data distribution of the target models in order to craft stronger perturbations.
4) We present comprehensive analysis of the proposed objective which includes: (a) a thorough comparison of our approach with the data-dependant counterparts, and (b) evaluation of the strength of UAPs in the presence of various defense mechanisms.

The rest of this paper is organized as follows: section 2 presents detailed account of related works, section 3 discusses the proposed data-free objective to craft image-agnostic adversarial perturbations, section 4 demonstrates the effectiveness of *GD-UAP* to craft UAPs across various tasks, section 5 hosts a thorough experimental analysis of *GD-UAP* and finally section 6 concludes the paper.

2 Related Works

Szegedy *et al.* [4] demonstrated that despite their superior recognition performance, neural networks are susceptible to adversarial perturbations. Subsequently, multiple other works [5], [6], [7], [16], [17], [18], [19] studied this interesting and surprising property of the machine learning models. Though it is first observed with recognition models, the adversarial behaviour is noticed with models trained on other tasks such as semantic segmentation [11], [20], object detection [20], pose estimation [21] and deep reinforcement learning tasks [22]. There exist multiple methods to craft these malicious perturbations for a given data sample. For recognition tasks, they range from performing simple gradient ascent [5] on cost function to solving complex optimizations ([4], [7], [23]). Simple and fast methods such as FGSM [5] find the gradient of loss function to determine the adversarial perturbation. An iterative version of this attack presented in [14] achieves better fooling via performing the gradient ascent multiple times. On the other hand, complex approaches such as [7] and [4] find minimal perturbation that can move the input across the learned classification boundary in order to flip the predicted label. More robust adversarial attacks have been proposed recently that transfer to real world [14] and are invariant to general image transformations [24].

Moreover, it is observed that the perturbations exhibit transferability, that is, perturbations crafted for one model can fool other models with different architectures and different training sets as well ([4], [5]). Further, Papernot *et al.* [13] introduced a practical attacking setup via model distillation to understand the *black-box* attack. Black-box attack assumes no information about the target model and its training data. They proposed to use a target model's substitute to craft the perturbations.

The common underlying aspect of all these techniques is that they are intrinsically data dependent. The perturbation is crafted for a given data sample independently of others. However, recent works by Moosavi-Dezfooli *et al.* [8] and Metzen *et al.* [11] showed the existence of input-agnostic

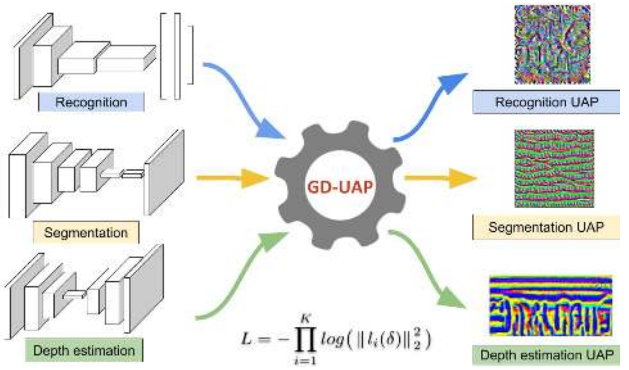


Fig. 1. Overview of the proposed generalized objective to craft "Image agnostic" (Universal) Adversarial Perturbations for a given target CNN. Input to our method is a task specific target CNN. The proposed objective, which is independent of the underlying task, then crafts UAPs without utilizing any data samples. The crafted UAPs are transferable to other models trained to perform the same underlying task as the target CNN.

perturbations that can fool the models over multiple images. In [8], authors proposed an iterative procedure based on Deepfool attacking [7] method to craft a universal perturbation to fool classification models. Similarly, in [11], authors craft universal perturbations that can affect target segmentation output. However, both these works optimize for different task specific objectives. Also, they require training data to craft the image-agnostic perturbations. Unlike the existing works, our proposed method *GD-UAP* presents a data-free objective that can craft perturbations without the need for any data samples. Additionally, we introduce a generic notion of fooling across multiple computer vision tasks via over-firing the neuron activations. Particularly, our objective is generalizable across various vision models in spite of differences in terms of architectures, regularizers, underlying tasks, etc.

3 Proposed Approach

First, we introduce the notation followed throughout the paper. $\mathcal{X}$ denotes the distribution of images in $\mathbb{R}^d$. f denotes the function learned by the CNN that maps an input image $x \sim \mathcal{X}$ to its output $f(x)$. δ denotes the image-agnostic perturbation learned by our objective. Similar to input x, δ also belongs to $\mathbb{R}^d$. Though the proposed objective is task independent, for ease of understanding we explain the proposed approach in the context of object recognition. Note that the proposed *objective* is generalizable across multiple vision tasks to craft *task specific*, image-agnostic adversarial perturbations. Insterestingly, these crafted task specific perturbations exhibit cross model generalizability. Figure 1 presents the proposed generalizable approach to learn task specific, image agnostic perturbations in data-free scenario.

3.1 Data-free objective for fooling

The objective of our paper is to craft an image-agnostic perturbation $\delta \in \mathbb{R}^d$ that fools the CNN f for images from the target distribution $\mathcal{X}$ without utilizing any samples from it. That is, we synthesize a δ such that

$$f(x+\delta) \neq f(x), \quad \text{for } x \sim \mathcal{X}. \tag{1}$$

The pixel intensities of δ are restricted by an imperceptibility constraint. Typically, it is realized as a max-norm constraint in terms of l_∞ or l_2 norms (e.g. [5], [8], [9], [11]). In this paper, for all our analysis we impose l_∞ norm. Thus, the aim is to find a δ such that

$$\begin{aligned} f(x+\delta) &\neq f(x), \quad \text{for } x \in \mathcal{X}; \\ \| \delta \|_\infty &< \xi. \end{aligned} \tag{2}$$

However, the focus of the proposed work is to craft δ without requiring any data samples. The data-free nature of our approach prohibits us from utilizing eqn. 2 for learning δ, as we do not have access to data x. Therefore, we instead propose to fool the CNN by contaminating the extracted representations of the input at multiple layers of the architecture. In other words, as opposed to the typical "flipping the label" objective, we attempt to "over-fire" the features extracted at multiple layers. That is, we craft a perturbation δ such that it leads to additional activation firing at each layer and thereby misleading the features (filters) at the following layer. The accumulated effect of the contamination eventually leads the CNN to misclassify.

The perturbation essentially causes filters at a particular layer to spuriously fire and extract inefficacious information. Note that in the presence of data (during attack), in order to mislead the activations from retaining useful discriminative information, the perturbation (δ) has to be highly effective. Also, the imperceptibility constraint (second part of eqn. 2) on δ makes it more challenging.

Hence without utilizing any data x, we seek an image-agnostic perturbation δ that can produce maximal spurious activations at each layer of a given CNN. In order to craft such a δ we start with a random perturbation and optimize for the following objective:

$$Loss = -\log\left(\prod_{i=1}^{K} \|l_i(\delta)\|_2\right), \quad \text{such that} \quad \| \delta \|_\infty < \xi. \tag{3}$$

where $l_i(\delta)$ is the activation in the output tensor (after the non-linearity) at layer i when δ is fed to the network f. K is the number of layers in f at which we maximize the activations caused by δ, and ξ is the max-norm limit on δ.

The proposed objective computes product of activation magnitude at all the individual layers. We observed product resulting in stronger δ than other forms of aggregation (e.g. sum). To avoid working with extreme values ($\approx$ 0), we apply log on the product. Note that the objective is open-ended as there is no optimum value to reach. We would ideally want δ to cause as much strong disturbance at all the layers as possible, within the imperceptibility constraint. More discussion on the motivation and working of the proposed objective is presented in Section 5.

3.2 Implementation Details

We begin with a target network f which is a trained CNN whose parameters are frozen and a random perturbation δ. We then perform the proposed optimization to update δ for causing strong activations at multiple layers in the given network. Typically, it is considered that the convolution $(conv)$ layers learn information-extracting features which are then classifed by a series of fc layers. Hence, we optimize our objective only at all the *conv* layers. This was empirically found to be more effective than optimizing at all layers as well. In case of advanced architectures such as GoogLeNet [25] and ResNet [26], we optimize at the last layers of all the inception (or residual) blocks and the independent *conv* layers. We observed that optimizing at these layers results in δ with a fooling capacity similar to the one resulting from optimizing at all the intermediate layers as well (including the *conv* layers within the inception/residual blocks). However, since optimizing at only the last layers of these blocks is more efficient, we perform the same.

Note that the optimization updates only the perturbation δ, not the network parameters. Additionally, no image data is involved in the optimization. We update δ with the gradients computed for loss in eqn. (3) iteratively till the fooling performance of the learned δ gets saturated on a set of validation images. In order to validate the fooling performance of the learned δ, we compose an unrelated substitute dataset (D). Since our objective is not to utilize data samples from the training dataset, we randomly select $1,000$ images from a substitute dataset to serve as validation images. It is a reasonable assumption for an attacker to have access to $1,000$ unrelated images. For crafting perturbations to object recognition models trained on ILSVRC dataset [27], we choose random samples from Pascal VOC-2012 [28] dataset. Similarly, for semantic segmentation models trained on Pascal VOC [28], [29], we choose validation samples from ILSVRC [27], for depth estimation models trained on KITTI dataset [30] we choose samples from Places-205 [31] dataset.

3.3 Exploiting additional priors

Though *GD-UAP* is a data-free optimization for crafting image-agnostic perturbations, it can exploit simple additional priors about the data distribution $\mathcal{X}$. In this section we demonstrate how *GD-UAP* can utilize simple priors such as (i) mean value and dynamic range of the input, and (ii) target data samples.

3.3.1 Mean and dynamic range of the input

Note that the proposed optimization (eqn. (3)) does not consider any information about $\mathcal{X}$. We present only the norm limited δ as input and maximize the resulting activations. Hence, during the optimization, input to the target CNN has a dynamic range of $[-\xi,\ \xi]$ $(\xi = 10)$. However, during the inference time, input lies in $[0,\ 255]$ range. Therefore, it becomes very challenging to learn perturbations that can affect the neuron activations in the presence of strong (an order higher) input signal x. Hence, in order to make the learning easier, we may provide this useful information about the data $(x \in \mathcal{X})$, and let the optimization better explore the space of perturbations. Thus, we slightly modify our objective to craft δ relative to the dynamic range of the data. We create pseudo data d via randomly sampling from a Gaussian distribution whose mean (μ) is equal to the mean of training data and variance (σ) is such that 99.9% of the samples lie in $[0,\ 255]$, the dynamic range of input. Thus, we solve for the following loss:

$$Loss = -\sum_{d\sim\mathcal{N}(\mu,\sigma)} \log\left(\prod_{i=1}^{K} \|l_i(d+\delta)\|_2\right), \quad \text{such that} \ \ \|\delta\|_\infty < \xi. \tag{4}$$

Essentially, we operate the proposed optimization in a subspace closer to the target data distribution $\mathcal{X}$. In other words, d in eqn. (4) acts as a place holder for the actual data and helps to learn perturbations which can over-fire the neuron activations in the presence of the actual data. A single Gaussian sample with twice the size of the input image is generated. Then, random crops from the Gaussian sample, augmented with simple techniques such as random cropping, blurring, and rotation are used for the optimization.

3.3.2 Target data samples

Now, we modify our data-free objective to utilize samples from the target distribution $\mathcal{X}$ and improve the fooling ability of the crafted perturbations. Note that in the case of data availability, we can design direct objectives such as reducing confidence for the predicted label or changing the predicted label, etc. However, we investigate if our data-free objective of over-firing the activations, though is not designed to utilize data, crafts better perturbations when data is presented to the optimization. Additionally, our objective does not utilize data to manipulate the predicted confidences or labels. Rather, the optimization benefits from prior information about the data distribution such as the dynamic range, local patterns, etc., which can be provided through the actual data samples. Therefore, with minimal data samples we solve for the following optimization problem

$$Loss = -\sum_{x\sim\mathcal{X}} \log\left(\prod_{i=1}^{K} \|l_i(x+\delta)\|_2\right), \quad \text{such that} \ \ \|\delta\|_\infty < \xi. \tag{5}$$

Presenting data samples to the optimization procedure is a natural extension to presenting the dynamic range of the target data alone (section 3.3.1). In this case, we utilize a subset of training images on which the target CNN models are trained (similar to [8], [11]).

3.4 Improved Optimization

In this subsection, we present improvements to the optimization process presented in our earlier work [9]. We observe that the proposed objective quickly accumulates δ beyond the imposed max-norm constraint (ξ). Because of the clipping performed after each iteration, the updates after δ reaches the constraint are futile . To tackle this saturation, δ is re-scaled to half of its dynamic range (i.e. $[-5,\ 5]$). Not only does the re-scale operation allow an improved utilization of the gradients, it also retains the pattern learnt in the optimization process till that iteration.

In our previous work [9], the re-scale operation is done in a regular time interval of 300 iterations. Though this re-scaling helps to learn better δ, it is inefficient since it performs blind re-scaling without verifying the scope for updating δ. This is specially harmful later in the learning process, when the perturbation may not be re-saturated in 300 iterations.

Therefore, we propose an adaptive re-scaling of δ based on the rate of saturation (reaching the extreme values of ± 10) in its pixel values. During the optimization, at each iteration we compute the proportion (p) of the pixels in δ that reached the max-norm limit ξ. As the learning progresses, more number of pixels reach the max-norm limit and because of the clipping, eventually get saturated at ξ. Hence, the rate of increase in p decreases as δ saturates. We compute the rate of saturation, denoted as S, of the pixels in δ after each iteration during the training. For consecutive iterations, if increase in p is not significant (less than a predefined threshold θ), we perform a re-scaling to half the dynamic range. We observe that this adaptive re-scaling consistently leads to better learning.

3.5 Algorithmic summarization

In this subsection, for the sake of brevity we summarize the proposed approach in the form of an algorithm. Algorithm 1 presents the proposed optimization as a series of steps. Note that it is a generic form comprising of all the three variations including both data-free and with prior versions.

For ease of reference, we repeat some of the notation. F_t is the fooling rate at iteration t, $l_i(x)$ is the activation caused at layer i of the CNN f for an input x, η is the learning rate used for training, Δ is the gradient of the loss with respect to the input δ, S_t is the rate of saturation of pixels in the perturbation δ at iteration t, θ is the threshold on the rate of saturation, F_t is the fooling rate, H is the patience interval of validation for verifying the convergence of the proposed optimization.

Algorithm 1: Algorithm summarizing our approach to craft image-agnostic adversarial perturbations via data-free objective and exploiting various data priors.

```
Data: Target CNN $f$, data $g$. Note that $g = 0$ for
      data-free case, $g = d \sim \mathcal{N}(\mu, \sigma)$ for range prior
      case, and $g = x$ for training data samples case.
Result: Image-agnostic adversarial perturbation $\delta$.
1  Randomly initialize $\delta_0 \sim \mathcal{U}[-\xi,\ \xi]$
2  $t = 0$
3  $F_t = 0$
4  do
5      $t \leftarrow t + 1$
6      Compute $l_i(g + \delta)$
7      Compute loss = $-\sum \log \left( \prod_{i=1}^{K} \|l_i(g + \delta)\|_2 \right)$
8      Update[1] $\delta_t : \delta_t \leftarrow \delta_{t-1} - \eta\Delta$
9      Compute the rate of saturation $S_t$ in the $\delta_t$ pixels
10     if $S_t < \theta$ then
11         $\delta_t \leftarrow \delta_t/2$
12     end
13     Compute $F_t$ of $\delta_t$ on substitute dataset $D$
14 while $F_t <$ min. of $\{F_{t-H}, F_{t-H+1} \dots F_{t-1}\}$;
15 $j \leftarrow$ argmax. of $\{F_{t-H}, F_{t-H+1} \dots F_{t-1}\}$
16 Return $\delta_j$
```

[1] Note that the generic update equation 8 is only representative and not the exact equation implemented.

3.6 Generalized Fooling Rate (GFR)

While the notion of 'fooling' has been well defined for the task of image recognition, for other tasks it is unclear. Hence, in order to provide an interpretable metric to measure 'fooling', we introduce Generalized Fooling Rate (*GFR*), making it independent of the task, and dependent on the metric being used for evaluating the model's performance.

Let M be a metric for measuring the performance of a model for any task, where the range of M is $[0, R]$. Let the metric take two inputs $\hat{y}$ and y, where $\hat{y}$ is the predicted output and y is the ground truth output, such that the performance of the model is measured as $M(\hat{y}, y)$. Let $\hat{y_\delta}$ be the output of the model when the input is perturbed with a perturbation δ. Then, the Generalized Fooling Rate with respect to measure M is defined as:

$$GFR(M) = \frac{R - M(\hat{y_\delta}, \hat{y})}{R}. \tag{6}$$

This definition of Generalized Fooling rate (*GFR*) has the following benefits:

- *GFR* is a natural extension of 'fooling rate' defined for image recognition, where the fooling rate can be written as $GFR(Top1) = 1 - Top1(\hat{y_\delta}, \hat{y})$, where $Top1$ is the Top-1 Accuracy metric.
- Fooling rate should be a measure of the change in model's output caused by the perturbation. Being independent of the ground truth y, and dependant only on $\hat{y_\delta}$ and $\hat{y}$, *GFR* primarily measures the change in the output. A poorly performing model which however is very robust to adversarial attacks will show very poor *GFR* values, highlighting its robustness.
- *GFR* measures the performance of a perturbation in terms of the damage caused to a model with respect to a metric. This is an important feature, as tasks such as depth estimation have multiple performance measures, where some perturbation might cause harm only to some of the metrics while leaving other metrics unaffected.

For all the tasks considered in this work, we report *GFR* with respect to a metric as a measure of 'fooling'.

4 GD-UAP: Effectiveness across tasks

In this section, we present the experimental evaluation to demonstrate the effectiveness of *GD-UAP*. We consider three different vision tasks to demonstrate the generalizability of our objective, namely, object recognition, semantic segmentation and unsupervised monocular depth estimation. Note that the set of applications include both classification and regression tasks. Also, it has both supervised and unsupervised learning setups, and various archtectures such as fully convolutional networks, and encoder-decoder networks. We explain each of the tasks separately in the following subsections.

For all the experiments, the ADAM [32] optimization algorithm is used with the learning rate of 0.1. The threshold θ for the rate of saturation S is set to 10^{-5} and ξ value of

TABLE 1
Fooling rates for *GD-UAP* perturbations learned for object recognition on ILSVRC dataset [27]. Each row of the table shows fooling rates for perturbation learned on a specific target model when attacking various other models (columns). These rates are obtained by *GD-UAP* objective with range prior (sec. 3.3.1). Diagonal rates indicate white-box attack scenario and off-diagonal ones represent black-box attack scenario.

Model	CaffeNet	VGG-F	GoogLeNet	VGG-16	VGG-19	Resnet-152
CaffeNet	**87.02**	65.97	49.40	50.46	49.92	38.57
VGG-F	59.89	**91.91**	52.24	51.65	50.63	**40.72**
GoogLeNet	44.70	46.09	**71.44**	37.95	37.90	34.56
VGG-16	50.05	55.66	46.59	**63.08**	56.04	36.84
VGG-19	49.11	53.45	40.90	55.73	**64.67**	35.81
Resnet-152	38.41	37.20	33.22	27.76	26.52	37.3

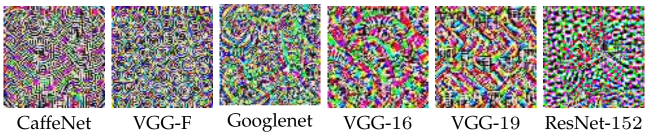


Fig. 2. Universal adversarial perturbations crafted by *GD-UAP* objective for multiple models trained on ILSVRC [27] dataset. Perturbations were crafted with $\xi = 10$ using the range prior (sec. 3.3.1).Images are best viewed in color.

10 is used. Validation fooling rate F_t is measured on the substitute dataset D after every 200 iterations only when the threshold of the rate of saturation is crossed. If it is not crossed, F_t is measured after every 400 iterations. Note that the algorithm specific hyper-parameters are **not** changed across tasks or across priors.

For all the following experiments, perturbation crafted with different priors are denoted as $P_{NP}, P_{RP},$ andP_{DP} for the *No prior, Range prior*, and *Data prior* scenario respectively. To emphasize the effectiveness of the proposed objective, we present fooling rates obtained by a random baseline perturbation. Since our learned perturbation is norm limited by ξ, we sample random δ from $\mathcal{U}[-\xi, \xi]$ and compute the fooling rates. In all tables in this section, the 'Baseline' column refers to this perturbation.

4.1 Object Recognition

We utilized models trained on ILSVRC [27] and Places-205 [31] datasets, viz. CaffeNet [33], VGG-F [34], Googlenet [25], VGG-16 [35], VGG-19 [35], ResNet-152 [26]. For all experiments, pretrained models are used whose weights are kept frozen throughout the optimization process.Also, in contrast to UAP [8], we do not use training data in the data-free scenario (sec. 3.1 and sec. 3.3.1). However, as explained earlier, we use $1,000$ images randomly chosen from Pascal VOC-2012 [28] training images as validation set (D in Algorithm 1) for our optimization. Also, in case of exploiting additional data prior (sec. 3.3.2), we use limited data from the corresponding training set. For the final evaluation on ILSVRC of the crafted UAPs, $50,000$ images from the validation set are used. Similarly, for Places-205 dataset, $20,500$ images from the validation set are used.

4.1.1 Fooling performance of the data-free objective

Table 1 presents the fooling rates achieved by our objective on various network architectures. Fooling rate is the percentage of test images for which our crafted perturbation δ successfully changed the predicted label. Using the terminology introduced in sec. 3.6, fooling rate can also be written as $GFR(Top1)$. Higher the fooling rate, greater is the perturbation's ability to fool and lesser is the the classifier's robustness. Fooling rates in Table 1 are obtained using the mean and dynamic range prior of the training distribution (sec. 3.3.1). Each row in the table indicates one target model employed in the learning process and the columns indicate various models attacked using the learned perturbations. The diagonal fooling rates indicate the *white-box attacking*, where all the information about the model is known to the attacker. The off-diagonal rates indicate *black-box attacking*, where no information about the model under attack is revealed to the attacker. However, the dataset over which both the models (target CNN and the CNN under attack) are trained is same. Our perturbations cause a mean white-box fooling rate of 69.24% and a mean black-box fooling rate of 45.13%. Given the data-free nature of the optimization, these fooling rates are alarmingly significant. The high fooling rates achieved by the proposed approach can adversely affect the real-world deploy-ability of these models.

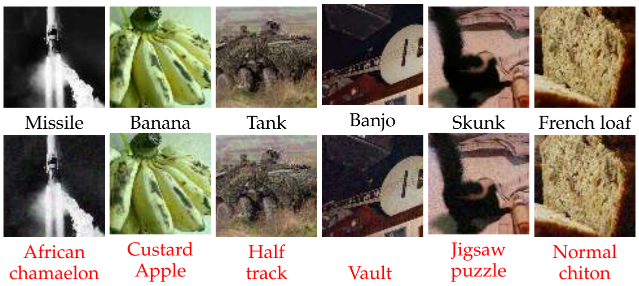


Fig. 3. Sample original and adversarial image pairs from ILSVRC validation set generated for VGG-19. First row shows original images and corresponding predicted labels, second row shows the corresponding perturbed images along with their predictions.

Figure 2 shows example image-agnostic perturbations (δ) crafted by the proposed method. Note that the perturbations look very different for each of the target CNNs. Interestingly, the perturbations corresponding to the VGG models look similar, which might be due to their architectural similarity. Figure 3 shows sample perturbed images ($x + \delta$) for VGG-19 [35] from ILSVRC [27] validation set. The top row shows the clean and bottom row shows the corresponding adversarial images. Note that the adversarially perturbed images are visually indistinguishable form their corresponding clean images. All the clean images shown in the figure are correctly classified and are successfully fooled by the added perturbation. Below each image, corresponding label predicted by the model is shown. Note that the correct labels are shown in black color and the wrong ones in red.

4.1.2 Exploiting the minimal prior

In this section, we present experimental results to demonstrate how our data-free objective can exploit the additional prior information about the target data distribution as discussed in section 3.3. Note that we consider two cases: (i) providing the mean and dynamic range of the data samples, denoted as range prior (sec. 3.3.1), and (ii) utilizing minimal data samples themselves during the optimization, denoted as data prior (3.3.2).

TABLE 2
Fooling rates for the proposed objective with and without utilizing prior information about the training data. For comparison, we provide the random baseline, existing data-free [9], and data dependent [8] objectives.

Model	Baseline	P_{NP}	P_{RP}	P_{DP}	FFF [9]	UAP [8]
CaffeNet	12.9	84.88	87.02	91.54	80.92	**93.1**
VGG-F	12.62	85.96	91.81	92.64	81.59	**93.8**
Googlenet	10.29	58.62	71.44	**83.54**	56.44	78.5
VGG-16	8.62	45.47	63.08	77.77	47.10	**77.8**
VGG-19	8.40	40.68	64.67	75.51	43.62	**80.8**
Resnet-152	8.99	29.78	37.3	66.68	-	**84.0**

Table 2 shows the fooling rates obtained with and without utilizing the prior information. Note that all the fooling rates are computed for white-box attacking scenario. For comparison, fooling rates obtained by our previous data-free objective [9] and a data dependent objective [8] are also presented. Important observations to draw from the table are listed below:

- Utilizing the prior information consistently improves the fooling ability of the crafted perturbations.
- A simple range prior can boost the fooling rates on an average by an absolute 10%, while still being data-free.
- Although the proposed objective is not designed to utilize the data, feeding the data samples results in an absolute 22% rise in the fooling rates. Due to this increase in performance, for all models (except ResNet-152) our method becomes comparable or even better than UAP [8], which is designed especially to utilizes data.

4.2 Semantic segmentation

In this subsection, we demonstrate the effectiveness of *GD-UAP* objective to craft universal adversarial perturbations for semantic segmentation. We consider four network architectures. The first two architectures are from FCN [36]: **FCN-Alex**, based on Alexnet [33], and **FCN-8s-VGG**, based on the 16-layer VGGNet [35]. The last two architectures are 16-layer VGGNet based **DL-VGG** [37], and **DL-RN101** [38], which is a multi-scale architecture based on ResNet-101 [26].

The FCN architetures are trained on Pascal VOC-2011 dataset [29], [39], consisting $9,610$ training samples and the remaining two architectures are trained on Pascal VOC-2012 dataset [28], [39], consisting $10,582$ training samples. However, for testing our perturbation's performance, we only use the validation set provided by [36], which consist of 736 images.

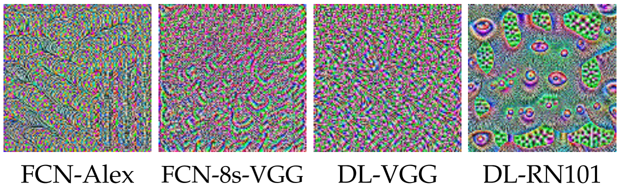


Fig. 4. Universal adversarial perturbations for semantic segmentation, crafted by the proposed *GD-UAP* objective for multiple models. Perturbations were crafted with “data w/ less BG” prior. Images are best viewed in color.

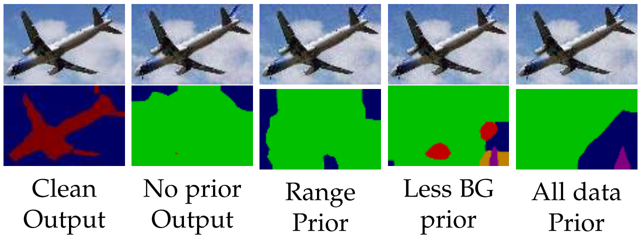


Fig. 5. Sample original and adversarial images from PASCAL-2011 dataset generated for **FCN-Alex**. First row shows clean and adversarial images with various priors. Second row shows the corresponding predicted segmentation maps.

Semantic segmentation is realized as assigning a label to each of the image pixels. That is, these models are typically trained to perform pixel level classification into one of 21 categories (including the background) using the cross-entropy loss. Performance is commonly measured in terms of mean IOU (intersection over union) computed between the predicted map and the ground truth. Extending the UAP generation framework provided in [8] to segmentation is a non-trivial task. However, our generalizable data-free algorithm can be applied for the task of semantic segmentation without any changes.

Similar to recognition setup, we present multiple scenarios for crafting the perturbations ranging from no data to utilizing data samples from the target distribution. An interesting observation with respect to the data samples from Pascal VOC-2012, is that, in the $10,582$ training samples, 65.4% of the pixels belong to the ‘background’ category. Due to this, when we craft perturbation using training data samples as target distribution prior, our optimization process encounters roughly 65% pixels belonging to ‘background‘ category, and only 35% pixels belonging to the rest 20 categories. As a result of this data imbalance, the perturbation is not sufficiently capable to corrupt the features of pixels belonging to the categories other than ‘background’. To handle this issue, we curate a smaller set of $2,833$ training samples from Pascal VOC-2012, where each sample has less than 50% pixels belonging to ‘background’ category. We denote this as “data w/ less BG”, and only 33.5% of pixels in this dataset belong to the ‘background’ category. The perturbations crafted using this dataset as target distribution prior show a higher capability to corrupt features of pixels belonging to the rest 20 categories. Since mean IOU is the average of IOU accross the 21 categories, we further observe that perturbations crafted using “data w/ less BG” cause a substantial reduction in the mean IOU measure as well.

Table 3 shows the generalized fooling rates with respect to the mean IOU ($GFR(mIOU)$) obtained by *GD-UAP* perturbations under various data priors. As explained in section 3.6, the generalized fooling rate measures the change in the performance of a network with respect to a given metric, which in our case is the mean IOU. Note that,

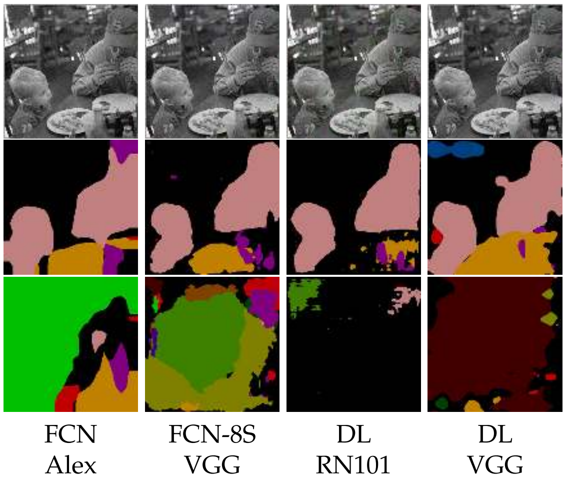


Fig. 6. Segmentation predictions of multiple models over a sample perturbed image. Perturbations were crafted using the "data w/ less BG" prior. The first row shows the perturbed input image, the second shows the segmentation output of clean sample image, and the third shows the segmentation output of the perturbed sample image. Images are best viewed in color.

TABLE 3
Generalized fooling rates achieved by the perturbations crafted by the proposed approach under various settings. Note that for comparison, fooling rates achieved by random perturbations are also presented.

Model	Baseline	No Data	Range Prior	All Data	Data W/ less BG
FCN-Alex	14.29	80.15	86.57	85.96	**89.61**
FCN-8s-VGG	9.24	49.42	55.04	61.15	**67.19**
DL-VGG	10.66	55.90	58.96	44.82	**66.68**
DL-RN101	8.8	37.06	35.6	**58.62**	56.61

similar to the recognition case, the fooling performance monotonically increases with the addition of data priors. This observation emphasizes that the proposed objective, though being an indirect, can rightly exploit the additional prior information about the training data distribution. Also, for all the models (Other than **DL-RN101**), "data w/ less BG" scenario results in the best fooling rate. This can be attributed to the fact that in "data w/ less BG" scenario we reduce the data-imbalance which in turn helps to craft perturbations that fool both background and object pixels.

In Table 4 we present the mean IOU metric obtained on the perturbed images learned under various scenarios along with original mean IOU obtained on clean images. It is clearly observed that the random perturbation (the baseline) is not effective in fooling the segmentation models. However, the proposed objective crafts perturbations within the same range that can significantly fool the models. We also show the mean IOU obtained by Xie *et al.* [20], an image specific adversarial perturbation crafting work. Note that since *GD-UAP* is an image-agnostic approach, it is unfair to expect similar performance as [20]. Further, the mean IOU shown by [20] for **DL-VGG** and **DL-RN101** models (bottom 2 rows of Table 4) denote the transfer performance, i.e., black-box attacking and hence show a smaller drop of the mean IOU from that of clean images. However, they are provide as an anchor point for evaluating image-agnostic perturbations generated using *GD-UAP*.

TABLE 4
Comparison of mean IOU obtained by various models against *GD-UAP* perturbations. Comparison with image specific adversaries [20] is also presented. ∗ denotes being image-specific and † denotes a transfer attack (black-box attacking). Being image specific, [20] (ICCV 2017) outperforms our perturbations, however, even our no data perturbations cause more drop in mIOU than their transfer perturbations.

Model	Original	Baseline	No Data	Range prior	All data	Data w/ less BG	[20]
FCN-Alex	46.21	45.72	15.35	10.37	10.64	8.03	3.98*
FCN-8s-VGG	65.49	64.34	42.78	39.08	33.61	28.05	4.02*
DL-VGG	62.10	61.13	36.91	35.41	44.90	27.41	43.96*†
DL-RN101	74.94	73.42	56.40	58.66	37.45	39.00	73.01*†

Figure 4 shows sample image-agnostic adversarial perturbations learned by our objective for semantic segmentation. In Figure 4, we show the perturbations learned with "data w/ less BG" prior for all the models. Similar to the recognition case, these perturbations look different across architectures. Figures 5 shows example image and predicted segmentation outputs by **FCN-Alex** model for perturbations crafted with various priors. Top row shows the clean and the perturbed images. Bottom row shows the predictions for the corresponding inputs. Further, the type of prior utilized to craft the perturbation is mentioned below the predictions. Crafted perturbations are clearly successful in misleading the model to predict inaccurate segmentation maps.

Figure 6 shows the effect of perturbation on multiple networks. It shows the output maps predicted by various models for the same input perturbed with corresponding δ learned with "data w/ less BG" prior. It is interesting to note from Figure 6 that for the same image, with UAPs crafted using the same prior, different networks can have very different outputs, even if their outputs for clean images are very similar.

4.3 Depth estimation

Recent works such as [40], [41], [42] show an increase in use of convolutional networks for regression-based computer vision task. A natural question to ask is whether they are as susceptible to universal adversarial attacks, as CNNs used for classification. In this section, by crafting UAPs for convolutional networks performing regression, we show that they are equally susceptible to universal adversarial attacks. To the best of our knowledge, we are the first to provide an algorithm for crafting universal adversarial attacks for convolutional networks performing regression,

Many recent works like [40], [43], [44] perform depth estimation using convolutional network. In [40], the authors introduce Monodepth, an encoder-decoder architecture which regresses the depth of given monocular input image. We craft UAP using *GD-UAP* algorithm for the two variants of Monodepth, **Monodepth-VGG** and **Monodepth-ResNet50**. The network is trained using KITTI dataset [30]. In its raw form, the dataset contains $42,382$ rectified stereo pairs from 61 scenes, with a typical image being 1242×375 pixels in size. We show results on the eigen split, introduced in [45], which consist of $23,488$ images for training and validation, and 697 images for test. We use

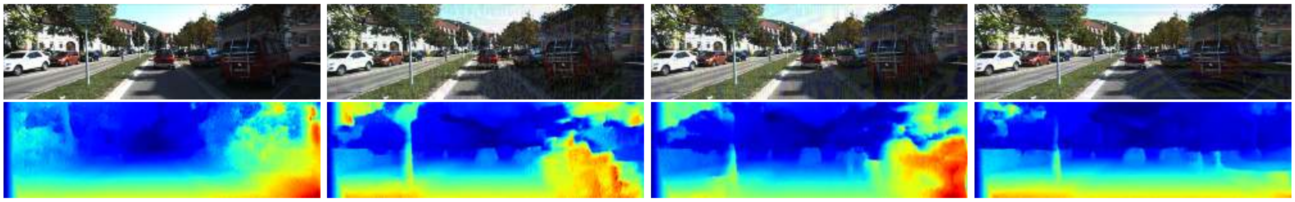


Fig. 7. Sample original and adversarial image pairs from KITTI dataset generated for **Monodepth-VGG**. First row shows clean and perturbed images with various priors. Second row shows the corresponding predicted depth maps.

TABLE 5
Performance of the crafted perturbations for **Monodepth-Resnet50** and **Monodepth-VGG** using various metrics for evaluating depth estimation on the eigen test-split of KITTI datset. Results are also presented for the clean data (Normal) and the train set mean. The best fooling results for each scenario are shown in bold. The evaluation of train-set mean performance has been taken from [40]. Note that the first four metrics are error based (higher means better fooling) and the later two are precision based (lower is better fooling).

Metrics	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta(1.25)$	$\delta(1.25^3)$
Monodepth-ResNet50						
Normal	0.133	1.148	5.549	0.230	0.829	0.970
Baseline	0.1339	1.1591	5.576	0.231	0.827	0.969
P_{NP}	0.201	1.810	6.603	0.352	0.688	0.908
P_{RP}	0.319	3.292	9.064	**0.640**	**0.460**	**0.717**
P_{DP}	**0.380**	**10.278**	**10.976**	0.402	0.708	0.900
Train-mean	0.361	4.826	8.102	0.377	0.638	0.894
Monodepth-VGG						
Normal	0.148	1.344	5.927	0.247	0.803	0.964
Baseline	0.149	1.353	5.949	0.248	0.800	0.963
P_{NP}	0.192	1.802	6.626	0.325	0.704	0.929
P_{RP}	0.212	2.073	6.994	**0.364**	**0.658**	**0.906**
P_{DP}	**0.355**	**9.612**	**10.592**	0.390	0.714	0.908
Train-mean	0.361	4.826	8.102	0.377	0.638	0.894

TABLE 6
GFR with respect to $\delta < 1.25$ metric for the task of depth estimation

Model	Baseline	No data	Range prior	Data prior
Monodepth-VGG	0.4%	15.3%	**22.7%**	21.3%
Monodepth-Resnet50	2%	21.3%	**47.6%**	24.3%

the same crop size as suggested by the authors of [45] and evaluate at the input image resolution.

As in the case of object recognition, UAPs crafted by the proposed method for monodepth also show the potential to exploit priors about the data distribution. We consider three cases, (i) providing no priors (P_{NP}), (ii)range prior (P_{RP}), and (ii) data prior (P_{DP}). For providing data priors, we randomly pick $10,000$ image samples from the KITTI train dataset. To attain complete independence of target data, we perform validation on a set of $1,000$ randomly picked images from Places-205 dataset. The optimization procedure followed is the same as in the case of the previous two task.

Table 5 show the performance of **Monodepth-Resnet50** and **Monodepth-VGG**under the presence of the various UAPs crafted by the proposed method. As can be observed from the table, the crafted UAPs have a strong impact on the performance of the network. For both the variants of monodepth, UAPs crafted with range prior, bring down the accuracy with the threshold of 1.25 units ($\delta < 1.25$) by 25.7% on an average. With data priors, the crafted UAPs are able to increase the Sq Rel (an error metric) to almost 10 times the original performance. Under the impact of the crafted UAPs, the network's performance drops below that of the depth-baseline (*Train-mean*), which uses the train set mean as the prediction for all image pixels. Figure 7 shows the input-output pair for **Monodepth-VGG**, where the input is perturbed by the various kinds of UAPs crafted.

Table 6 shows the Generalized Fooling Rates (GFR) with respect $\delta < 1.25$, i.e. $GFR(\delta < 1.25)$. It is observed that P_{RP} has higher $GFR(\delta < 1.25)$ than P_{DP}. This may appear as an anomaly as P_{DP}, which has access to more information, should cause stronger harm to the network than P_{RP}. This is indeed reflected in terms of multiple metrics such as $Abs.Rel.Error$, and $RMSE$ (ref. Table 5). In fact, P_{DP} is able to reduce these metrics even below the values achieved by the train-set mean. This clearly shows that P_{DP} is indeed stronger than P_{RP} (in terms of these metrics).

However, in terms of the other metrics, such as $\delta < 1.25$ and $\delta < 1.25^3$, it is observed that P_{RP} causes more harm. These metrics measure the $\%$ of pixels where $f(x) - G(x)$ (where $G(x)$ represents the ground truth depth at x) is lesser than pre-defined limits. In contrast, metrics such as $Abs.Rel.Error$, and $RMSE$ measure the overall error of the output. Hence, based on the effect of P_{DP} and P_{RP} on these metrics, we infer that while P_{DP} shifts fewer pixels than P_{RP}, it severely shifts those pixels. In fact, as shown in Figure 7, it is often noticed that P_{DP} causes the network to completely miss some nearby objects, and hence predicting very high depth at such locations, whereas P_{RP} causes a anomalous estimation at a higher number of pixels.

The above situation shows that the 'fooling' performance of a perturbation can vary based on the metric used for analysis. Further, conclusions based on a single metric may only partially reflect the truth. This motivated us to propose $GFR(m)$, a metric dependant measurement of 'fooling', which clearly indicates the metric dependence of 'fooling'.

5 *GD-UAP*: Analysis and Discussion

In this section, we provide additional analysis of *GD-UAP* on various fronts. First, we clearly highlight the multiple advantages of *GD-UAP* by comparing it with other approaches. In the next subsection we provide a thorough experimental evaluation of image-agnostic perturbation in the presence of various defence mechanism. Finally, we end this

TABLE 7
Comparison of data-free objectives. Fooling rates achieved by maximizing l_2 norm (*GD-UAP*) vs. mean activation ([9]) when utilizing data samples.

Model	FFF [9]	l_2 GD-UAP
CaffeNet	88.35	**91.54**
VGG-16	72.68	**77.77**
Resnet-152	65.43	**66.68**

TABLE 8
Effect of data dependency on crafting the perturbations. Data dependent objectives [8] suffer significant drop in fooling ability when arbitrary data samples are utilized for crafting. A→ B denotes that data A is used to craft perturbations to fool models trained on data B. Note that fooling rates for our approach are crafted without utilizing any data samples (denoted with *).

Model	Places-205 → ILSVRC		ILSVRC → Places-205	
	Ours	UAP [8]	Ours	UAP [8]
CaffeNet	**87.02***	73.09	**88.61***	77.21
GoogLenet	**71.44***	28.17	**83.37***	52.53

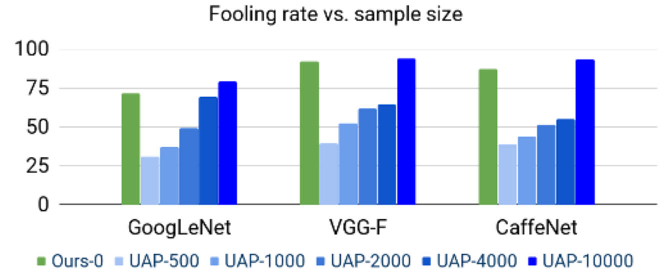


Fig. 8. Reliance of the data dependent objective UAP [8] on the size of available training data samples. Note that our approach utilizes no data samples and achieves competitive fooling performance.

TABLE 9
We present the fooling rate comparison of the existing data dependent UAP[8] approach and the proposed approach.

Priors	Ours	UAP[8]
No data	58.62	No convergence
Range Prior	71.44	10.56
Data Prior	83.54	78.5

section with a discussion on how *GD-UAP* perturbations work.

5.1 Comparison with other approaches

5.1.1 Comparison of data-free objective

First, we compare the effectiveness of *GD-UAP* against the existing data-free objective proposed in [9]. Specifically, we compare maximizing the mean versus l_2 norm (energy) of the activations caused by the perturbation δ (or $x/d + \delta$ in case of exploiting the additional priors).

Table 7 shows the comparison of fooling rates obtained with both the objectives (separately) in the improved optimization setup (3.4). We have chosen 3 representative models across various generations of models (CaffeNet, VGG and ResNet) to compare the effectiveness of the proposed objective. Note that the improved objective consistently outperforms the previous one by a significant 3.18%. Similar behaviour is observed for other vision tasks also.

5.1.2 Data dependent vs. Data-free objectives

Now, we demonstrate the necessity of data dependent objective [8] to have samples from the target distribution only. That is, methods (such as [8]) that craft perturbations with fooling objective (i.e. move samples across the classification boundaries) require samples from only the training data distribution during the optimization. We show that crafting with arbitrary data samples leads to significantly inferior fooling performance.

Table 8 shows the fooling rates of data dependent objective [8] when non-target data samples are utilized in place of target samples. Experiment in which we use Places-205 data to craft perturbations for models trained on ILSVRC is denoted as Places-205 → ILSVRC and vice versa. For both the setups, a set of $10,000$ training images are used. Note that, the rates for the proposed method are obtained without utilizing any data (with range prior) and rates for data-free scenario can be found in Table 2. Clearly the fooling rates for UAP [8] suffer significantly, as their perturbations are strongly tied to the target data. On the other hand, for the proposed method, since it does not craft via optimizing a fooling objective, the fooling performance does not decrease. Importantly, these experiments show that the data dependent objectives are not effective when samples from the target distribution are not available. This is a major drawback as it is difficult to procure the training data in practical scenarios.

Additionally, as the data dependent objectives rely on the available training data, the ability of the crafted perturbations heavily depends on the size of the available training data. We show that the fooling performance of UAP [8] significantly decreases as the size of the available samples decreases. Figure 8 shows the fooling rates obtained by the perturbations crafted for multiple recognition models trained on ILSVRC by UAP [8] with varying size of samples available for optimization. We craft different UAP [8] perturbations (using the codes provided by the authors) utilizing only $500, 1,000, 2,000, 4,000$ and $10,000$ data samples and evaluate their ability to fool various models. The performance of the crafted perturbations decreases drastically (shown in different shades of blue) as the available data samples are decreased during the optimization. For comparison, fooling rates obtained by the proposed data-free objective is shown in green.

5.1.3 Capacity to expoit minimal priors

An interesting question to ask is whether the data dependant approach UAP [8] can also utilize minimal data priors. To answer this, we craft perturbations using the algorithm presented in UAP, with different data priors (including no data priors). Table 9 presents the comparison of *GD-UAP* against UAP [8] with different priors. The numbers are computed on the ILSVRC validation set for Googlenet. We observe that the UAP algorithm does not even converge in the absence of data, failing to create a data-free UAP. When range prior (Gaussian noise in the data range) is used to craft, the resulting perturbations demonstrate a very low fooling rate of 10.56 compared to 71.44 of the proposed method. This is not surprising, as a significant decrease in the performance of UAP can be observed even due to a simple mismatch of training and target data (ref Table 8). Finally, when actual data samples are used for crafting, UAP [8] achieves 78.5 success rate which is closer to 83.54 of the proposed method.

From these results, we infer that since UAP [8] is a data dependent method, it requires corresponding data samples to craft effective perturbations. When noise samples are presented for optimization, the resulting perturbations fail to generalize to the actual data samples. In contrast, *GD-UAP* solves for an activation objective and exploits the prior as available.

5.2 Robustness of UAPs Against Defense Mechanisms

While many recent works propose novel approaches to craft UAPs, the current literature does not contain a thorough analysis of UAPs in the presence of various defense mechanisms. If simple defense techniques could render them harmless, they may not present severe threat to deployment of deep models. In this subsection, we investigate the strength of *GD-UAP* against various defense techniques. Particularly, we consider (1) *Input Transformation Defenses*, such as Gaussian blurring, Image quilting, etc., (2) *Targeted Defense Against UAPs*, such as Perturbation Rectifying Networks (PRN) [46] and (3) *Defense through robust architectural design* such as scattering networks [47].

We evaluate the performance of UAPs generated from *GD-UAP*, as well as data-dependent approach UAP [8] against various defense mechanisms.

TABLE 10
The fooling rates computed for various UAP algorithms for under different defenses on GoogLeNet. Acc_{top1} represents the Top-1 Accuracy on clean images in the presence of defense mechanism.

Model	Acc_{top1}	Data-Independant		Data-Dependant	
		P_{NP}	P_{RP}	P_{DP}	UAP [8]
No Defense	69.74	58.62	71.44	**83.54**	78.5
Input transformation defenses					
10-Crop	70.94	46.6	54.3	**66.6**	-
Gaussian Smoothing	58.12	25.80	35.62	32.66	**32.78**
Median Smoothing	51.78	35.08	**46.98**	43.96	37.02
Bilateral Smoothing	60.28	17.06	21.50	22.34	**22.74**
Bit-Reduction (3-bit)	63.20	48.66	61.34	**72.30**	69.62
Bit-Reduction (2-bit)	45.50	46.74	56.76	60.50	65.40
JPEG (75%)	67.50	35.66	51.22	**61.38**	42.62
JPEG (50%)	64.16	29.34	41.84	**44.62**	31.40
TV Minimization	50.50	27.9	31.9	**30.9**	26.22
Image Quilting	51.30	30.76	**36.12**	34.80	25.84
Targeted Defenses Against UAPs					
PRN Defense	68.90	31.34	46.60	**52.50**	21.35

5.2.1 Input Transformation Defenses

For defense by input transformations, inline with [48] and [49], we consider the following simple defenses: (1) 10-Crop Evaluation, (2) Gaussian Blurring, (3) Median Smoothing, (4) Bilateral Filtering (5) JPEG Compression, and (6) Bit-Depth Reduction. Further, we evaluate two sophisticated image transformations proposed in [50], namely, (7) TV-Minimization and (8) Image Quilting (Using code provided by the authors).

Table 10 presents our experimental evaluation of the defenses on the Googlenet. As we can see, while the fooling rate of all UAPs is reduced by the defenses (significantly in some cases), it is achieved at the cost of model's accuracy on clean images. If Image Quilting, or TV-normalization is used as a defense, it is essential to train the network on quilted images, without which, a severe drop in accuracy is observed. However, in majority of the defenses, UAPs flip labels for more than 45% of the images, which indicates threat to deployment. Further, as *Bilateral Filtering* and *JPEG compression* show strong defense capability at low cost to accuracy, we evaluate the performance of our GD-UAP perturbations (with range prior) on 6 classification networks in the presence of these two defenses. This is shown in Table 11. We note that when the defense mechanism significantly lowers the fooling rates, a huge price is paid in terms of % drop in Top-1 Accuracy (D_{Acc}), which is unacceptable. This further indicates the poor fit of input transformations as a viable defense.

5.2.2 Targeted Defense Against UAPs

Now, we turn towards defenses which have been specifically engineered for UAPs. We evaluate the performance of the various UAPs, with the Perturbation Rectifying Network(PRN) [51] as a defense. PRN is trained using multiple UAPs generated from the algorithm presented in UAP [8] to rectify perturbed images. In Table 10, we present the fooling rates obtained on Googlenet using various perturbations, using the codes provided by the authors of [51]. While PRN is able to defend against [8], it shows very poor performance against our data prior perturbation. This is due to the fact that PRN is trained using UAPs generated from UAP [8] only, indicating that PRN lacks generalizability to input-agnostic perturbations generated from other approaches. Furthermore, it is also observed that various simple input transformation defenses outperform PRN. Hence, while PRN is an important step towards defenses against UAPs, in its current form, it provides scarce security against UAPs from methods it is not trained on.

5.2.3 Defense through robust architectural design

As our optimization process relies on maximizing $||f(x + \delta)||_2$ by increasing $\Pi||l_i(x+\delta)||_2 \forall i \in \{1, 2, ...\}$ (where $l_i(.)$ represents the input to layer l_{i+1}), one defense against our attack can be to train the network such that the change in the output to a layer l_i minimally effects the output of the network $f(.)$. One method for achieving this target can be to minimize the Lipschitz Constant K_i of each linear transformation in the network. As K_i controls the upper bound of the value of $||f(x+\delta) - f(x)||_2$ with respect to $||l_i(x+\delta) - l_i(x)||_2$, minimizing $K_i \forall i \in \{1, 2, ...\}$ can lead

TABLE 11
The effect of strong input defenses on perturbations crafted from our objective with range-prior. D_{Acc} represents the percentage drop in Top-1 Accuracy on clean images.

Model	Caffenet		VGG-F		GoogLeNet		VGG-16		VGG-19		ResNet-152	
	$\%D_{Acc}$	FR	$\%D_{Acc}$	FR	$\%D_{Acc}$	FR	$\%D_{Acc}$	FR	$\%D_{Acc}$	FR	$\%D_{Acc}$	FR
No Defense	-	87.1	-	91.8	-	71.4	-	63.1	-	64.7	-	37.3
JPEG(50%)	5.23%	72.1	0.5%	77.5	8%	41.8	2.6%	49.2	2.4%	64.7	4.7%	37.3
Bilateral	15.2%	34.8	18.2%	34.8	13.5%	21.5	14.7%	35.8	14.2%	28.2	9.7%	25.4

TABLE 12
Fooling Rate for Hybrid-network [47], ResNet-18 and VGG-13 by black-box attack.

	No-Prior	Range-Prior	Data-Prior
		Perturbation from VGG-13	
Hybrid-network	**28.91**	**30.96**	**33.90**
ResNet-18	35.08	39.14	43.92
		Perturbation from ResNet-18	
Hybrid-network	**25.30**	**27.82**	**39.36**
VGG-13	37.82	44.70	60.86

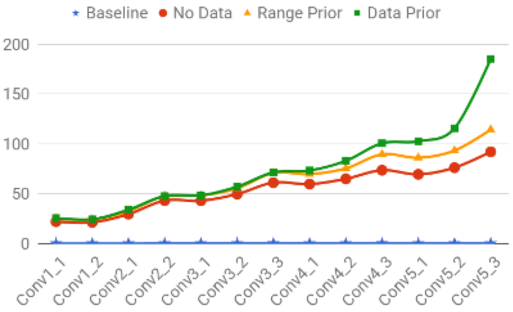


Fig. 9. Percentage relative change in the extracted representations caused by our crafted perturbations at multiple layers of VGG-16.

to a stable system where minor variation in input layer do not translate to high variation in output. This would translate to low fooling rate when attacked by adversarial perturbations.

In [52], Bruna *et al.* introduce scattering network. This network consist of scattering transform, which linearize the output deformation with respect to small input deformation, ensuring that the Lipschitz constant is ≤ 1. In [47], a hybrid approach was proposed, which uses scattering transforms in the initial layers, and learn convolutional layers (Res-blocks, in specific) on the transformed output of the scattering layers, making scattering transform based approaches feasible for Imagenet. The proposed Hybrid-network gave performance comparable to ResNet-18 and VGG-13 while containing much lesser layers.

We now evaluate the fooling rate that *GD-UAP* perturbations achieve in the Hybrid-Network, and compare it to fooling rate achieved on VGG-13 and ResNet-18 networks. In the Hybrid-network, ideally we would like to maximize $||l_i(x+\delta)||_2$ at each of the Res-Block output. However, as $\partial S(x)/\partial x$, where $S(.)$ represents the Scattering Transform, is non-trivial, we only perform black-box attacks on all the networks.

Table 12 shows the results of the black-box attack on Hybrid Networks. Though Hybrid networks on an average decrease the fooling rate by 13% when compared to other models, they still remain vulnerable.

5.3 Analyzing how *GD-UAP* works

As demonstrated by our experimentation in section 4, it is evident that *GD-UAP* is able to craft highly effective perturbations for a variety of computer vision tasks. This highlights an important question about the Convolutional Neural Networks (CNNs):"How stable are the learned representations at each layer?" That is, Are the features learned by the CNNs robust to small changes in the input? As mentioned earlier 'fooling' refers to instability of the CNN in terms of its output, independent of the task at hand.

We depict this as a stability issue with the learned representations by CNNs. We attempt to learn the optimal perturbation in the input space that can cause maximal change in the output of the network. We achieve this via learning perturbations that can result in maximal change in the activations at all the intermediate layers of the architecture. As an example, we consider VGG-16 CNN trained for object recognition to illustrate the working of our objective. Figure 9 shows the percentage relative change in the feature activations ($\frac{||l_i(x+\delta)-l_i(x)||_2 \times 100}{||l_i(x)||_2}$) at various layers in the architecture. The percentage relative change in the feature activations due to the addition of the learned perturbation increases monotonically as we go deeper in the network. Because of this accumulated perturbation in the projection of the input, our learned adversaries are able to fool the CNNs independent of the task at hand. This phenomenon explains the fooling achieved by our objective. We can also observe that with the utilization of the data priors, the relative perturbation further increases which results in better fooling when the prior information is provided during the learning.

Our data-free approach, consists of increasing $||l_i(\delta)||$ across the layers, to increase $f(\delta)$. As Figure 9 shows, this crafts a perturbation, which leads to an increase in $f(x+\delta)$. One may intrepret this increase to be caused due to the locally linear nature of CNNs. i.e., $f(x+\delta) \approx f(x) + f(\delta)$. However, our experiments reveal that the feature extractor f might not be locally linear. We observe the relation between the quantities $||f(x+\delta) - f(x)||_2$ and $||f(\delta)||_2$, where $||.||_2$ represents the L_2-norm, and $f(.)$ represents the output of the last convolution layer of the network. Figure 10 presents the comparison of these two quantities for VGG-16 computed during the proposed optimization with no prior.

TABLE 13
Relative Shift in Classification Layer's input vs. fooling rate for **VGG-16** for object recogntiion. The relative shift has been evaluation on 1000 random images from ILSVRC validation set, while the fooling rate is evaluated on the entire ILSVRC validation set.

Perturbation	Rel. Shift in input to fc_8 (classification) layer	Fooling rate
Baseline	0.0006	8.62
No Prior	0.867	45.47
Range Prior	1.142	63.08
All data Prior	3.169	77.77

From the observations, we infer: (1) $||f(x+\delta)-f(x)||_2$ is **not** approximately equal to $||f(\delta)||_2$, and hence, f is **not** observed to be locally linear, (2) However, $||f(x+\delta)-f(x)||_2$ is strongly correlated to $||f(\delta)||_2$, and our data-free optimization approach exploits this correlation between the two quantities. To summarize, our data-free optimization exploits the correlation between the quantities, $||f(x+\delta)-f(x)||_2$ and $||f(\delta)||_2$, rather than the local-linearity of feature extractor f.

Finally, the relative change caused by our perturbations at the input to classification layer (fc_8 or softmax) can be clearly related to the fooling rates achieved for various perturbations. Table 13 shows the relative shift in the feature activations ($\frac{||l_i(x+\delta)-l_i(x)||_2}{||l_i(x)||_2}$) that are input to the classification layer and the corresponding fooling rates for various perturbations. Note that they are highly correlated, which explains why the proposed objective can fool the CNNs trained across multiple vision tasks.

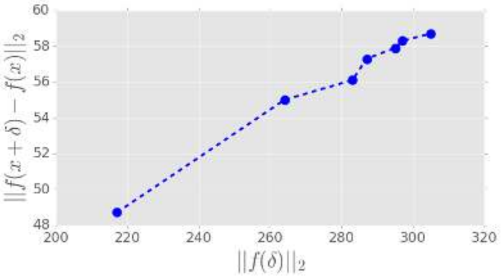


Fig. 10. Correlation between $||f(x+\delta)-f(x)||_2$ and $||f(\delta)||_2$ computed for VGG-16 model. Plot shows the fit between $||f(x+\delta)-f(x)||_2$ and $||f(\delta)||_2$ computed during the training at iterations just before the δ gets saturated.

6 Conclusion

In this paper, we have proposed a novel data-free objective to craft image-agnostic (universal) adversarial perturbations (UAP). More importantly, we show that the proposed objective is generalizable not only across multiple CNN architectures but also across diverse computer vision tasks. We demonstrated that our seemingly simple objective of injecting maximal "adversarial" energy into the learned representations (subject to the imperceptibility constraint) is effective to fool both the classification and regression models. Significant transfer performances achieved by our crafted perturbations can pose substantial threat to the deep learned systems in terms of black-box attacking.

Further, we show that our objective can exploit minimal priors about the target data distribution to craft stronger perturbations. For example, providing simple information such as the mean and dynamic range of the images to the proposed objective would craft significantly stronger perturbations. Though the proposed objective is data-free in nature, it can craft stronger perturbations when data is utilized.

More importantly, we introduced the idea of generalizable objectives to craft image-agnostic perturbations. It is already established that the representations learned by deep models are susceptible. On top of it, the existence of generic objectives to fool "any" learning based vision model independent of the underlying task can pose critical concerns about the model deployment. Therefore, it is an important research direction to be focused on in order to build reliable machine learning based systems.

References

[1] B. Biggio, G. Fumera, and F. Roli, "Pattern recognition systems under attack: Design issues and research challenges," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 07, 2014.

[2] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013, pp. 387–402.

[3] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, ser. AISec '11, 2011.

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

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

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

[7] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] K. R. Mopuri, U. Garg, and R. V. Babu, "Fast feature fool: A data independent approach to universal adversarial perturbations," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[10] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," *arXiv preprint arXiv:1703.08603*, 2017.

[11] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer, "Universal adversarial perturbations against semantic image segmentation," in *International Conference on Computer Vision (ICCV)*, 2017.

[12] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, Jan. 2009.

[13] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples," *arXiv preprint arXiv:1602.02697*, 2016.

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

[15] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. L. Yuille, "Adversarial examples for semantic segmentation and object detection," *arXiv preprint arXiv:1703.08603*, 2017.

[16] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," *arXiv preprint arXiv:1502.02590*, 2015.

[17] A. Fawzi, S. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: from adversarial to random noise," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

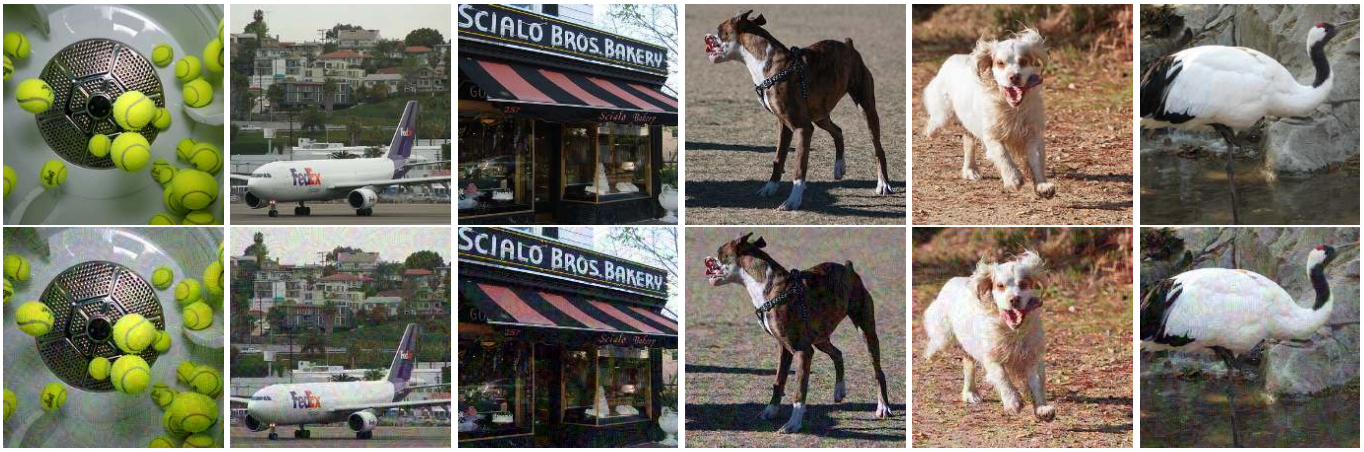


Fig. 11. Sample failure case for the object recognition using VGG-16 model. Top row shows multiple clean images from ILSVRC validation set. Bottom row shows the adversarial images generated by adding the perturbation crafted utilizing the no data prior. Note that for all the shown images the perturbation fails to change the predicted label.

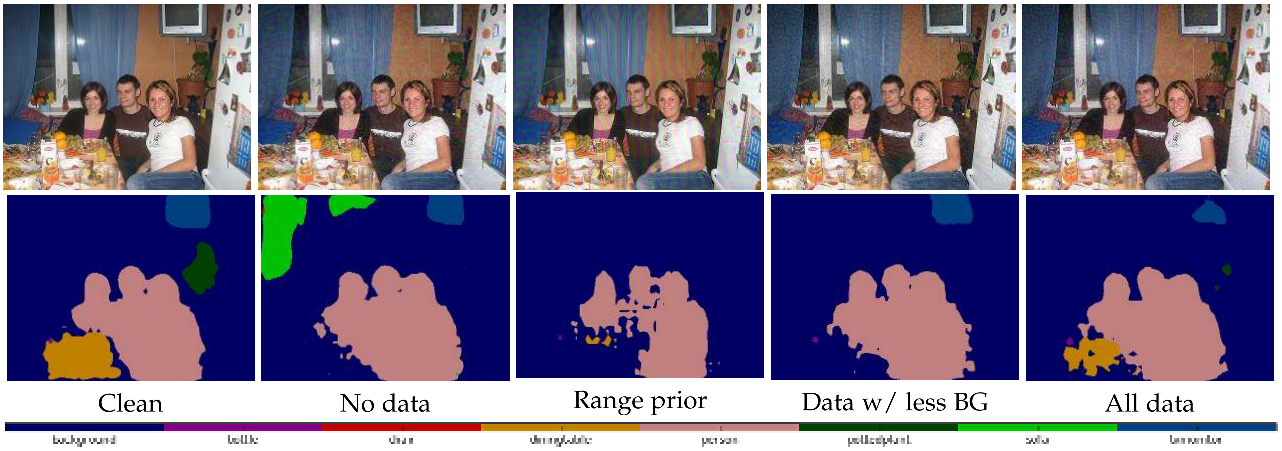


Fig. 12. Sample failure case for the semantic segmentation using the FCN-8s-VGG model. Top row shows the clean and corresponding perturbed images for no prior to various priors. Bottom row shows the predicted segmentation maps. Note that the people segments are undisturbed by the addition of various perturbations.

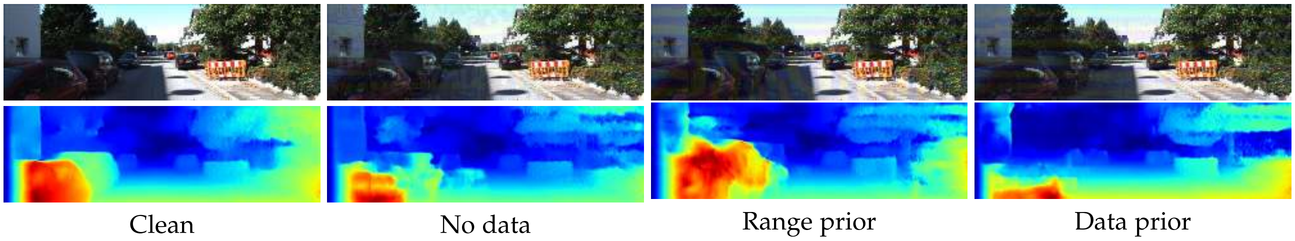


Fig. 13. Sample failure case for the depth estimation using Monodepth-VGG model. Note that, top row shows clean image and the corresponding perturbed images with no data case and various prior cases. Bottom row shows the corresponding depth predictions.

[18] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[19] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2016, pp. 25–32.

[20] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *International Conference on Computer Vision (ICCV)*, 2017.

[21] M. M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in *Advances in Neural Information Processing Systems*, 2017, pp. 6980–6990.

[22] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," *arXiv preprint arXiv:1701:04143*, 2017.

[23] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, "Measuring neural net robustness with constraints," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[24] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," *arXiv preprint arXiv:1707.07397*, 2017.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov,

D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *arXiv preprint arXiv:1409.4842*, 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

[27] 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 (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.”

[29] M. Everingham, L. VanGool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results.”

[30] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[31] B. Zhou, A. Khosla, À. Lapedriza, A. Torralba, and A. Oliva, “Places: An image database for deep scene understanding,” *arXiv preprint arXiv:1610.02055*, 2016.

[32] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint: arXiv:1412.6980*, 2014.

[33] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

[34] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *Proceedings of the British Machine Vision Conference (BMVC)*, 2014.

[35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:abs/1409.1556*, 2014.

[36] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, vol. 39, no. 4, 2017.

[37] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” in *International Conference on Learning Representations (ICLR)*, 2015.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *arXiv preprint arXiv:1606.00915*, 2016.

[39] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, “Semantic contours from inverse detectors,” in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[40] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] D. B. Sam, S. Surya, and R. V. Babu, “Switching convolutional neural network for crowd counting,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[42] A. Kendall and R. Cipolla, “Geometric loss functions for camera pose regression with deep learning,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6555–6564.

[43] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, “Semantic scene completion from a single depth image,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 190–198.

[44] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, “Multi-scale continuous crfs as sequential deep networks for monocular depth estimation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 161–169.

[45] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2366–2374.

[46] N. Akhtar, J. Liu, and A. Mian, “Defense against universal adversarial perturbations,” *arXiv preprint arXiv:1711.05929*, 2017.

[47] E. Oyallon, E. Belilovsky, and S. Zagoruyko, “Scaling the Scattering Transform: Deep Hybrid Networks,” in *International Conference on Computer Vision (ICCV)*, 2017.

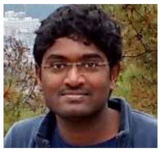
[48] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” *arXiv preprint arXiv:1704.01155*, 2017.

[49] 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.

[50] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, “Countering adversarial images using input transformations,” *arXiv preprint arXiv:1711.00117*, 2017.

[51] N. Akhtar, J. Liu, and A. Mian, “Defense against universal adversarial perturbations,” *arXiv preprint arXiv:1711.05929*, 2017.

[52] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.



Konda Reddy Mopuri received an M.Tech degree from the IIT Kharagpur, India in 2011. He is currently pursuing Ph.D. degree with the Department of Computational and Data Sciences, Indian Institute of Science (IISc), Bangalore, India, advised by Prof. R. Venkatesh Babu. He worked in Samsung India, Bangalore from 2011 to 2012. His research interests include computer vision, and machine learning with an emphasis on studying the deep learned visual representations.

Aditya Ganeshan is a project assistant at Video Analytics Lab, Indian Institute of Science, Bangalore. He received his Integrated Master of Science in Applied Mathematics from Indian Institute of Technology, Roorkee. His research Interest includes machine learning, reinforcement learning and functional analysis.

R. Venkatesh Babu received the Ph.D. degree from the Department of Electrical Engineering, Indian Institute of Science (IISc), Bangalore, India. He held post-doctoral position with the Norwegian University of Science and Technology, Norway, and IRISA/INRIA, Rennes, France. He was a Research Fellow with Nanyang Technological University, Singapore. Currently, he is an Associate Professor at Dept. of Computational and Data Sciences, IISc. His research interests span signal processing, compression, machine vision, image/video processing, machine learning, and multimedia.