IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Adversarial Attack using Sparse Representation of Feature Maps

MAHAM JAHANGIR[1], FAISAL SHAFAIT[1,2].
[1]School of Electrical Engineering and Computer Science, National University of Sciences & Technology (NUST), Islamabad, Pakistan.
[2]Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad, Pakistan

Corresponding author: Maham Jahangir(e-mail: mjahangir.phdcs17seecs@seecs.edu.pk).

ABSTRACT Deep neural networks can be fooled by small imperceptible perturbations called adversarial examples. Although these examples are carefully crafted, they involve two major concerns. In some cases, adversarial examples generated are much larger than minimal adversarial perturbations while in others the attack method involves an extensive number of iterations making it infeasible. Moreover, the sparse attacks are either too complex or are not sparse enough to achieve imperceptibility. Therefore, attacks designed should be fast and minimum in terms of ℓ_2-norm. In this research, we used a dictionary learning technique to generate sparse adversarial examples based on feature maps of target images. We present two novel algorithms to tune the dictionary learning process and feature map selection. The results on MNIST and Imagenet show our attack is better or competitive with the state-of-the-art methods. We also compared our method with sparse attacks recently introduced in literature. As a result, we have achieved comparable attack success rate when compared to the state-of-the-art with smaller ℓ_2-norm. We also tested the efficacy of our attack in the presence of defense mechanisms and none of the defenses were able to combat the effect of our proposed attack

INDEX TERMS adversarial attacks, dictionary learning, sparse representation

I. INTRODUCTION

DEEP Neural Networks have gained a lot of success and reached human-level performance in image recognition, detecting faces and objects, autonomous driving, reading addresses, solving captchas, and many more [1], [2]. The convolutional neural networks particularly have been useful since 2012, after giving promising results on Imagenet Large Scale Visual Recognition Challenge [3]. Since that time improvements from researchers are coming at a high pace in the form of a wide range of applications, more complex and deep architectures, and improving the overall classification process.

Despite the success of CNN on image recognition tasks, we still lack in complete understanding of these complex networks. Szegedy et al. [4] explored the unusual mistake that deep networks-based classifiers can make. They can be fooled by carefully computed images called adversarial images, revealing the unstable nature of these architectures. These images are indistinguishable from humans when compared to the original images.

This area has received a lot of interest from researchers and practitioners all over the world. One stream of research focuses on generating adversarial attacks with the lowest imperceptibility while the other focuses on creating defenses for such attacks. The researchers are still working on the precise inner workings and reasoning of deep networks. The attacks help understand the internal working of these architectures and thus motivate extensive research on designing robust classifiers. For this purpose, a lot of attacks have been introduced by different researchers in the literature.

Fast Gradient Sign Method [5] and C&W [6] are among the famous state-of-the-art attack methods. The current mainstream possesses certain problems: In terms of ℓ_2-norm distortions, the C&W is argued to be the most effective attack but is slow since it requires thousands of iterations making it unsuitable for adversarial training too [7]. Researchers have argued perturbations estimated using the FGSM are much larger than minimal adversarial perturbations [8]. Adversarial examples generated by iterative attacks contain a certain amount of redundant noises that cannot be completely removed by simply increasing the number of iterations [9]. In light of the above mentioned problems, the attacks designed should be fast, and minimum in terms of ℓ_2-norm.

Exploiting the internal details of DNNs to generate effective

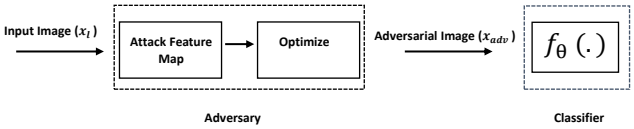

FIGURE 1. Block Diagram of our approach: it depicts a classifier being attacked by an adversary which perturbs the input x_l by adding feature map of target image

imperceptible attack is particularly relevant and the subject of this paper. In this paper, we addressed the above-mentioned problems using ideas from Sparse Representation, Sparse Coding, and Dictionary Learning. Sparse representation is a linear internal representation of images using only a few active coefficients making it easy to interpret and manipulate content-based image indexing and retrieval. This field uses a dictionary and a sparse linear combination of the atoms in the dictionary to represent every input signal. The computation of the representation coefficients X also remains a non-trivial operation which is solved by the Orthogonal Matching Pursuit (OMP) which is greedy and has a fast running time. It has received great interest in machine learning, pattern recognition, signal processing [10], and has been successfully applied to image classification [11], image compression [12], reconstruction, noise reduction [13], face recognition [14] etc.

Recently, some of the nominal work that focuses on sparse attacks include Corner Search, Sparse Fool and Greedy Fool. All these methods are either suffer from high complexity that they cannot be extended to high-resolution images or perturbs redundant pixels therefore, not applicable to real scenarios. Current algorithms are highly complexed NP-hard problems. The adversarial examples generated by these models usually consist of high-magnitude noise, concentrated over a small number of pixels. As a result, the adversarial images become quite perceptible and might even exceed the dynamic range of the image. We have tried to address the limitations of state-of-the-art methods mentioned above as well as recent sparse methods in this paper. The idea is to mimic the internal representation of target images. For this purpose, we designed an attack based on the feature maps from the first layer of convolutional neural networks. The perturbation designed using feature maps is added to the original image to attack the classifier as shown in Fig. 1. The Block Diagram of our approach shows a classifier being attacked by an adversary which perturbs the input x_l by adding a feature map of the target image. We have optimized our perturbation vector using dictionary learning to have a linear, non-redundant, sparse noise added to the original input image. Feature maps get the important pixels of a respective image that are used for classification.

Experiments on MNIST and Imagenet datasets show the efficacy of the proposed approach in terms of decreased error and smaller ℓ_2-norm even for a one-shot method. The proposed approach has been applied to both targeted and un-targeted scenarios.

We have also tested our adversarial images against various defense methods. The attack is not defended by any of the defense strategies.

We summarize our contributions as follows:

1) We used ideas from dictionary learning and sparse coding to generate adversarial attacks. These are the first attacks based on dictionary learning proposed so far, to the best of our knowledge.
2) We have tried to overcome the limitations of both state-of-the-art methods as well as recent sparse attacks.
3) We have also presented novel algorithms to learn tuned dictionary based on feature maps. These ideas to tune dictionaries can be extended to other machine learning problems solved by dictionary learning
4) We presented a comprehensive experimental analysis to back our approach. A detailed investigation on tuning the dictionary to create an effective attack and then testing it against various defense methods to prove its efficacy.
5) We motivate a new area for designing adversarial attacks.

The structure of the paper is as follows. The related literature is discussed in Section II. The detailed methodology is described in Section III followed by the experimental setting and details in Section IV. The discussion and analysis of results are tabulated in Section V. Finally, the experiments regarding defense strategies are explained and analyzed in Section VI and the paper is concluded in Section VII.

II. RELATED WORK

The vulnerability of neural networks towards adversarial examples was introduced by authors in [4]. The attacks can be targeted or un-targeted. In targeted attacks, the adversary forces the classifier into predicting a specified label, while any label in case of untargeted.

Among the state-of-the-art in [5] the authors' proposed Fast Gradient Sign Method which creates adversarial examples by computing the sign of the gradient of the loss of the input images. Later, iterative methods such as Deep Fool [15]

and C&W attacks [6] were introduced. C&W attacks are considered very strong and effective against defensive distillation. Universal adversarial attacks were also proposed early on to fool all kinds of neural networks [16]. Recently, steganographic universal adversarial perturbations are introduced by [17]. They used a single secret image (computed in the transform domain) to fool deep architectures. Similarly, Yahya et al. generated an adversarial attack by selecting a targeted watermark, using a steganographic approach [18]

As our work is related to internal representation as well as sparse attacks, the remaining matter of this section discusses the relationships to both areas. In [19] the authors used DNN logits as vectors to represent features and exploited them to create targeted universal attacks. These perturbations generalize well across different neural networks. They can be designed using the information of networks like training data, weights, etc called white-box attacks, or can be black-box in nature without knowing the architecture, learned weights, or training data. Moreover, these attacks are transferable among different architectures, and a lot of recent literature provides insights into the transferability of adversarial images [20], [21]. Moreover, authors are motivated to generate attacks that explain the deep representations of the model rather than fooling it [22]. Shi et. al [23], recently explained robustness through an adaptive iterative attack.

More recently, feature maps are used to generate transferable attacks. In [21] the source image is perturbed by reducing the distance between layer L activations of a source image and a target image in a white-box setting. The images are then fed into the black box model to test the transferability. Yucheng Shi et al. [9] argued in their research that there is no refinement mechanism to squeeze redundant noises in most of the attacks. Thus, their work is based on adding diversity by using gradient ascent and descent and then optimizing by filtering out noises of groups of similar pixels.

The other area of related work is the sparse representation and sparse attacks. Sparse attacks have been recently introduced in the field of adversarial attacks. Some of the early sparse attacks in adversarial setting includes JSMA [24], Sparse Fool [25], Corner Search [26] and Greedy Fool [27]. Sparse Fool [25] disrupts the geometrical properties of the images whereas, Corner Search [26] aims at minimizing the distance of the perturbation to the original image.

All these attacks have certain limitations: JSMA [24] is highly complex and is difficult to apply to high-resolution images. SparseFool [25] cannot perform a targeted attack and isn't sparse enough. In PGD [28] the number of pixels to be perturbed is defined beforehand therefore, it results in perturbing redundant pixels and might not be flexible for real scenarios. [29]

These researches present interesting ideas but are addressing different problems. We have used ideas from Sparse Representation and Dictionary Learning. Sparse Representation has gained a lot of attention in computer vision applications. It has wide applications in image reconstruction, denoising, image inpainting, and many more. Aharon et al. [30] proposed the K-SVD method to learn the dictionary to achieve sparse representation. As compared to previous sparse attacks, our work is different as we aim to discover a dictionary that can optimize the perturbation vector to achieve performance in terms of smaller ℓ_2-norm. The smaller ℓ_2-norm helps achieve imperceptibility one of the major limitations of existing work as highlighted above. We used dictionary learning to add sparse perturbation in input images which change the minimum important pixels of the clean image, another limitation highlighted above. It has been proved through various experiments that Dictionary Learning was able to overcome the limitations of existing work. In this paper, we learned the dictionary to optimize the targeted noises. Our dictionary consists of perturbations instead of clean images. This is the first time dictionary learning has been used for this task. However, the sparse representation has only been used as a defense mechanism to reduce feature space against adversarial attacks, to the best of our knowledge [31]–[33].

III. METHODOLOGY

In this section, we describe in detail the methodology of the proposed approach to generate adversarial images. We have introduced a novel dictionary learning technique that is based on the feature maps of the image associated with the targeted label. These sparse representations of feature maps serve as noise to be added to the original image.

We first formulate the problem in Section III-A. The methodology for sparse adversarial image generation is then divided into three phases. The tuned dictionary learning algorithm based on feature maps is explained in Section III-B. The computation of the perturbation vector using feature maps of the target image is explained in Section III-C. Finally, in the third phase, we generate an adversarial image from sparse perturbation vectors by the one-shot method explained in Section III-D. The detailed methodology highlighting all phases is illustrated in Fig. 2.

A. PROBLEM FORMULATION

Let X be the image space and Y be the label space. $f_\theta(.) : X \rightarrow Y$ is a classifier parameterized by θ that assigns a label y to an input image x. Let x_l denote the legitimate image to be perturbed by noise p. We aim to generate an adversarial example $x_a = x_l + p$ which is imperceptible from x_l but fools the classifier i.e. :

$$d(x_a, x_l) < \epsilon \;\; s.t. \;\; f_\theta(x_a) \neq y_l \tag{1}$$

d(.,.) is the distance e.g. ℓ_2-norm of the difference between the clean and the adversarial sample, y_l is the correct label of the legitimate input image, and ϵ is the perturbation scale which is often set to a very small value to ensure imperceptibility between x_a and x_l. In case of targeted attacks :

$$f_\theta(x_a) = y_t \tag{2}$$

where y_t is the target label we want the classifier to predict. In this work, we consider both targeted and un-targeted labels. We aim to inject the noise p to make a strong attack by

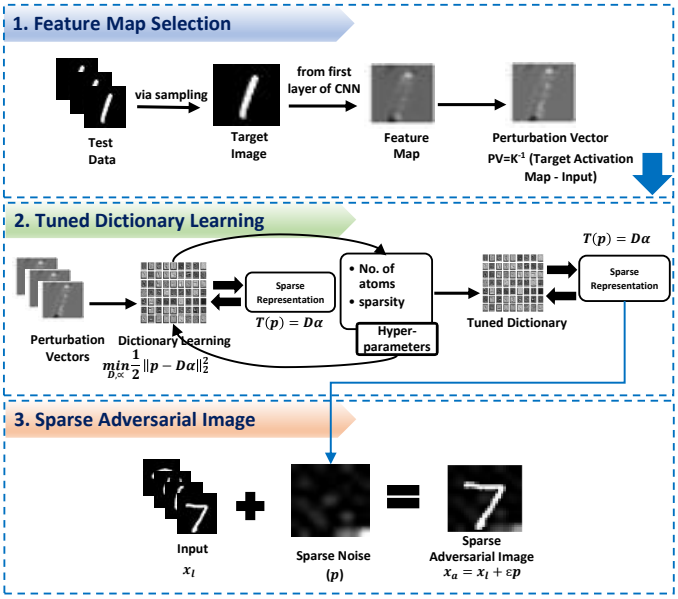

FIGURE 2. Top Row (Feature Map Selection):The perturbation vector $K^{-1}(\text{Target Feature Map} - \text{Input})$ is generated from the feature map of the target image. Middle Row (Tuned Dictionary Learning): The sparse image is generated by optimizing the perturbation vector through dictionary learning.Bottom Row(Sparse Adversarial Image): The sparse noise is added to the original input to generate an adversarial image.

learning an adverse transformation $T(.)$ such that adversarial detection-based defense methods should not be able to detect the attack. The noise is derived from the internal representation of the image x_t associated with the targeted label y_t. This enables the adversary to create an attack with a smaller norm as opposed to most of the attacks in the literature. Moreover, most detection-based defense mechanisms detect the attack based on the redundant noises left by the adversarial attacks. Therefore, it is desirable to make this transformation $T(.)$ stronger by preserving the important information required while limiting the space of adversarial noise. It should further remove redundant noises and should be difficult to detect by defense mechanisms.

B. TUNED DICTIONARY LEARNING ALGORITHM

The operator $T(p)$ transforms the perturbation vector derived from feature maps in close proximity to the local neighborhood of the image by linear projection. Let p be the perturbation vector, x_t the image associated with the target label y_t, we look for the transformation operator $T(.)$ satisfying the following conditions:

$$f_\theta(x_l + T(p)) = y_t \;\; s.t. \;\; d(x_l + T(p), x_t) \;\; < \epsilon \quad (3)$$

The classifier f assigns the targeted label to the fabricated input image which is our ultimate goal, given the condition that $T(p)$ (Transformed feature map of the target) and x_t (image associated with the target label) should be situated closely. We present the tuned dictionary learning algorithm to learn this transformation satisfying both conditions. We propose a feature map-based dictionary learning algorithm to learn this transformation. The idea is to mimic the internal representation of the target image. The image associated with the target label which we want the classifier to predict. So, we want to learn the transformation that should be close to the target image. For that purpose, we used the feature maps of the target image to create perturbation. Sparse representation approximates an input signal X by a sparse linear combination of items from an overcomplete dictionary. Let the projection of p be $T(p)$ given by:

$$T(p) = D\alpha \quad (4)$$

The projection in our algorithm is learned through a dictionary by the following optimization problem [10]. The optimization problem solved is a dictionary learning with an ℓ_1 penalty on the components.

$$\begin{aligned} &\min_{D,\alpha} \frac{1}{2} \parallel p - D\alpha \parallel_2^2 + \lambda \parallel \alpha \parallel_1 \\ &s.t. \parallel D_k \parallel_2 = 1 \;\; \forall \;\; k \in [0, n] \end{aligned} \quad (5)$$

where, p = perturbation signal and λ is a regularization parameter, and n is the number of dictionary atoms.The sparsity-inducing ℓ_1-norm also prevents learning components from noise when few training samples are available. The degree of penalization that is sparsity level can be adjusted through the α. Small values result in gently regularized coefficients, while larger values shrink many coefficients to zero. The squared error between the original and transformed

signal is the basis of tuning the dictionary learning algorithm. Unlike other dictionary learning algorithms that are used to learn a dictionary of clean images to support the application of denoising, compression, or inpainting, we learn the dictionary to optimize the targeted noises. Our dictionary is called an adverse dictionary as it consists of perturbations instead of clean images. We used sampled feature map selection to improve computational efficiency. We use a novel feature map selection technique to learn this dictionary which is explained in the preceding section. The performance of the dictionary learning algorithm is enhanced by tuning it after different hyper-parameter selection. The dictionary learning algorithm tunes these hyper-parameters based on the squared error. The algorithm runs for a fixed number of iterations with different values of hyper-parameters: sparsity level and the number of components. Our experiments show that the squared error is highly correlated with the selection of these hyper-parameters. We chose test set images from the MNIST dataset to conduct these experiments. These hyper-parameters and their effect is described later in the section on ablation studies. The detailed algorithm for dictionary learning is provided in Algorithm 1.

Algorithm 1: Tuned Dictionary Learning

Input: $P \rightarrow$ Set of all perturbation vectors;
Result: D $\rightarrow$ Dictionary
$Err \rightarrow$ Squared Error between the transformed image and the original target image ;
$T(p) \rightarrow$ Transformed perturbation learned through the dictionary of a single perturbation vector;
$x_t \rightarrow$ Target Image;
$N \rightarrow$ No. of iterations ;
$D_c \rightarrow$ Current Dictionary ;
$D \rightarrow$ Initial Dictionary ;
$k \rightarrow$ Sparsity ;
$n \rightarrow$ no. of atoms ;
$Err = \frac{1}{N}\sum_{n=1}^{N} \parallel T(p) - x_t \parallel_2$
$D = \min_{D,\alpha} \frac{1}{2} \parallel p - D\alpha \parallel_2^2 + \lambda \parallel \alpha \parallel_1$
$s.t. \parallel D_k \parallel_2 = 1 \ \ \forall \ \ k \in [0, n]$;
for $i < N$ **do**
  $D_c = \min_{D,\alpha} \frac{1}{2} \parallel p - D\alpha \parallel_2^2 + \lambda \parallel \alpha \parallel_1$
  $s.t. \parallel D_k \parallel_2 = 1 \ \ \forall \ \ k \in [0, n]$;
  if $Err(D) > Err(D_c)$ **then**
    $D = D_c$;
  update n;
  update k;
Return D

C. FEATURE MAP SELECTION TO LEARN THE DICTIONARY

In this paper, we aim to mimic the internal representations of the target inputs to create our adversarial images. The idea is to produce an adversarial image whose internal representation matches that of the target input. Sabour et al. tried to do the same by reducing the Euclidean distance between the source and the target guide images [34].

The internal representation is captured by using the feature maps of the target images. These feature maps result in an output of one filter applied to the previous layer. These filters also known as kernels, are called feature identifiers. The feature maps detect low-level features at initial layers of the CNN and high-level features as we go deep in the architecture. The low-level features are closely related to images, the high-level are difficult to map to the image. Therefore, we use feature maps from the first layer of the CNN.

The knowledge about feature maps and kernels to mimic the internal representation highlighted so far is used in this paper to generate the perturbation/noises for our adversarial images. The natural workflow of the CNN applies a kernel on an image to produce a feature map. We want to add noise in that image to generate a targeted feature map (feature map of the image we are targeting). The idea can be mathematically written as

$$K(x_l + p) = F_t \tag{6}$$

Therefore, the perturbation vector is given by

$$p = K^{-1}(F_t) - x_l \tag{7}$$

Here, x_l is the legitimate source image, and F_t is the feature map of the target image x_t. F_t is the feature map of a target image generated by a well-trained network. K is the pre-learned filter/kernel from the same well-trained network. K^{-1} is the deconvolution operation. The effect of deconvolution of CNN layers is discussed in detail in [35]. p is the perturbation we want to compute. The sparse representation of this perturbation will be added in the original image as noise described in detail in Section III-D. This perturbation is generated using the test data keeping the essence of a black-box attack where the adversary doesn't have access to the training data. We feed these perturbations to learn the dictionary for sparse representation.

Next, we explain a novel efficient feature map selection algorithm to improve dictionary learning. Feature Maps possess information about the important pixels of the image [21]. Likewise, learning a discriminative dictionary is necessary to improve representation. The traditional approaches often suffer from the problem of local minima. Therefore, researchers have proposed to learn dictionaries with good representational power, and better discrimination capabilities for all classes [13]. Therefore, the idea is to build a dictionary by selecting important and diverse inputs. We select important and diverse patches by greedily sampling the test data. The target image of a particular class is selected for a dictionary if the ℓ_2-norm of that image is greater than a threshold. This threshold is basically, the mean ℓ_2-norm of all the images in a particular class. This way we get to learn the dictionary with diverse images. The images of all classes are included and we try to include as many diverse images of the same class as possible. The detailed algorithm is presented in

Algorithm 2. The testing sets are used to sample feature maps. These selected images are then used to generate feature maps described earlier in this section. In this way, we learn a discriminative dictionary and optimize its performance by reducing the size of the number of atoms with sampling.

Algorithm 2: Feature Map Selection to Learn Dictionary

Result: $p \rightarrow$ Selected Feature Map as Perturbation
Input: $F \rightarrow$ Feature Map ;
$S \rightarrow$ Testing samples of the selected class ;
$K \rightarrow$ Pre-defined kernel ;
$x_l \rightarrow$ legitimate source input image;
$H \rightarrow$ threshold on ℓ_2-norm;
$H \leftarrow \frac{1}{N}\sum_{n=1}^{N} \parallel s_i \parallel_2$;
for $i < |S|$ **do**
  if $\parallel s_i \parallel_2 > H$ **then**
    $p \leftarrow K^{-1}(F_i) - x_l$;
  else
    $i \leftarrow i + 1$;
  Return p

D. SPARSE ADVERSARIAL IMAGE GENERATION

The sparse representation of images has gained growing interest. In this report, we solve our problem of redundant noises, and smaller ℓ_2-norm by feature map selection-based dictionary learning. We describe how a sparse representation framework has been tailored to generate sparse adversarial images. Since all the required pieces are together we finally generate adversarial images by adding the desired perturbation vector to the legitimate image controlled by ϵ given in (6). The ϵ determines the magnitude of noise to be added to the legitimate source image to maintain imperceptibility and limit the ℓ_2-norm of the adversarial image. The final noise is not a combination of different noises. We propose a feature map-based dictionary learning algorithm to learn this transformation. Sparse representation approximates an input signal X by a sparse linear combination of items from an overcomplete dictionary. The projection of p given by $T(p) = D\alpha$, it is mentioned that since we are adding this transformed (sparse representation) as noise to the original image so it's a dictionary of noises. The ℓ_2-norm is the calculated difference between the legitimate and the adversarial image.

$$x_a = x_l + \epsilon p \tag{8}$$

The final adversarial image is then fed to the classifier.

IV. EXPERIMENTAL SETTINGS AND RESULTS

We evaluated the proposed attack methodology for both targeted and un-targeted scenarios on MNIST (black and white handwritten digits) and Imagenet dataset (colored images). The sparse adversarial attacks are compared with the state-of-the-art attacks i.e. C&W [6] Corner Search [26] Sparse Fool [25], Greedy Fool [27] and FGSM [5]. C&W [6] is considered to generate adversarial examples with minimum ℓ_2 noise, yet it is impractical because of its high number of iterations [7], [8].

A. METRICS

In this section, we describe various metrics to define the performance of our algorithm. We report the mean and median ℓ_2-norm using the following formulae

$$d(x, x_a) = \parallel x - x_a \parallel_2 \tag{9}$$

$$\text{median} = \text{median}(d(x, x_a) \mid x \in X) \tag{10}$$

$$\text{average} = \frac{1}{N}\sum_{n=1}^{N} d(x, x_a) \mid x \in X \tag{11}$$

A smaller ℓ_2-norm distance indicates a stronger attack effect and higher transferability [9]. In the ablation, studies section targeted success rate (TSR) is calculated. The targeted success rate is the rate at which sparse adversarial images generated are classified as the target label. The larger the targeted success rate, the more effective the targeted attack. Another metric used in the ablation studies section is the Squared Error distance calculated between the transformed image and the image associated with the target label given as: $\parallel T(p) - x_t \parallel_2$
In the defense evaluation section fooling ratio is recorded for all the defense strategies. It is the percentage of images on which the classifier changes its prediction label after they are perturbed. The high values of the fooling ratio mean that the attacks are more strong. In this paper, it is shown that even after applying various defense strategies the fooling ratio of our proposed attack remains high.

B. MNIST

The training set consists of 50,000 images whereas, the test set consists of 10,000 images with resolution (28x28). The proposed attacks are evaluated on MNIST using a model with 99.25% Top-1 accuracy and an error of 0.04. We trained the model for 50 epochs with a learning rate of 0.01 using ADAM optimizer. The model was trained on a simple CNN architecture consisting of 6 layers. First, starting with 2 convolutional layers with 3x3 kernel size, then 2D max pooling size (2x2), followed by dropout (0.25), and finally Flatten and Dense Layers. The total trainable parameters were 55,658 We generated adversarial images using the proposed strategy with $\epsilon = 0.01$ for un-targeted and targeted attacks. The un-targeted images are poisoned with any perturbation vector. The 10,000 images from MNIST test data are all used for evaluation purposes. The experiments are conducted for the proposed approach as well as state-of-the-art attacks: FGSM, Corner Search, and C&W. The adversarial robustness toolbox is used to conduct experiments for FGSM and C&W [36]. The publicly available original code of corner search was used to conduct the experiments. The $\epsilon = 1$ is used for FGSM

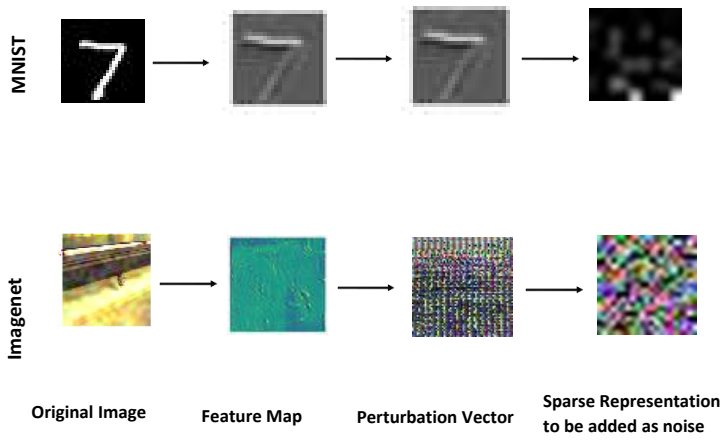

FIGURE 3. Left to Right: Original Image, Feature Map, Perturbation Vector, and finally Sparse Representation of perturbation vector to be used as adversarial noise generated through our approach for MNIST and Imagenet datasets.

TABLE 1. The classifier's loss on test data, attack success rate, mean and median ℓ_2-norm of the proposed attack compared with the state-of-the-art attacks on MNIST & Imagenet in un-targeted Scenario. A smaller ℓ_2-norm indicates a stronger attack.

MNIST					
Attack	**Loss on Test**	**Attack Succ.**	**Mean ℓ_2**	**Median ℓ_2**	**Run-time sec(s) (10K-images)**
FGSM	10.41	73%	0.1	0.1	557
Corner Search	2.09	88%	7.9	8.8	225100
CW	2.87	43%	0.01	0.01	4303
Ours	9.26	86%	0.1	0.1	245
Imagenet					
Attack	**Loss on Test**	**Attack Succ.**	**Mean ℓ_2**	**Median ℓ_2**	**Run-time sec(s) (1K-images)**
FGSM	2.5	58%	0.3	0.3	1240
CW	1.2	42%	0.0004	0.0004	42400
Sparse Fool	7.49	100%	6.5	5.4	144000
Greedy Fool	8.59	99%	1.02	0.89	1677
Ours	2.47	53%	0.02	0.02	631

TABLE 2. The error of the targeted attack, attack success rate, mean and median ℓ_2 of proposed attack compared with the state-of-the-art attacks on MNIST and Imagenet in targeted Scenario. A low value of the loss indicates a stronger targeted attack.

MNIST (Average Case)				
Attack	**Loss on Test**	**Attack Succ.**	**Mean ℓ_2**	**Median ℓ_2**
FGSM	8.7	18.3%	0.1	0.1
Corner Search	46.69	1.2%	0.7	0
CW	29.51	23%	0.005	0.005
Ours	5.6	17%	0.1	0.1
Imagenet (Average Case)				
Attack	**Loss on Test**	**Attack Succ.**	**Mean ℓ_2**	**Median ℓ_2**
FGSM	23.6	1%	1.36	1.36
CW	18.62	1%	0.001	0.001
Ours	18.62	1%	0.17	0.002

for un-targeted and targeted attacks. We did not use the same values of ϵ for FGSM because for smaller values (as used in our case) the method cannot attack the network at all. Results are reported both for targeted and un-targeted scenarios. For targeted attacks following the methodology from [7] we generate adversarial images for all classes of MNIST. This indicated 9 attacks per image. The results are reported by averaging overall attacks. The error, attack success rate, mean ℓ_2-norm, and median ℓ_2-norm are reported in every case.

C. IMAGENET

The Imagenet consists of (224x224) sized images from 1000 categories. The proposed attacks are evaluated on Imagenet using a pre-trained VGG-19 model with 70.2% Top-1 accuracy and an error of 1.20. The adversarial images for targeted attacks are created with $\epsilon = 0.0001$. The un-targeted are generated with $\epsilon = 0.0001$ with any sparse perturbation. We chose 1000 images from its validation set representing each category of class for evaluation purposes. The experiments are conducted for the proposed approach as well as state-of-the-art attacks: FGSM, C&W, SparseFool and GreedyFool. The experiments for state-of-the-art attacks are conducted using the library [36] for FGSM and C&W. The publicly available original implementations of SparseFool and GreedyFool were used to conduct the experiments. Results are reported both for targeted and un-targeted scenarios. For targeted attacks, following the methodology from [7], we generate adversarial images for 10 classes chosen randomly. The results are reported by averaging overall attacks. The $\epsilon = 0.01$ is used for un-targeted and $\epsilon = 0.9$ targeted attacks

while conducting experiments for FGSM. The error, attack success, mean ℓ_2-norm, and median ℓ_2-norm are reported in every case. We couldn't conduct experiments of corner search on Imagenet due to a lack of memory resources. It required 113 GiB for an array with shape (100352, 224, 224, 3). SparseFool-based attacks cannot be extended to targeted attacks, Therefore targeted attacks were not applicable in this case.

D. EXPERIMENTAL RESULTS

We report the mean, and median ℓ_2-norm using the formulae described above. A smaller ℓ_2-norm distance indicates a stronger attack effect.

1) Un-Targeted Attack

The results indicate that the proposed attack is effective in terms of ℓ_2-norm when compared to others. Table 1shows the performance of the un-targeted proposed attack on MNIST and Imagenet. The mean and median values of 0.1 for MNIST and 0.02 for Imagenet are calculated which is comparable to the state-of-the-art attacks in case of un-targeted attacks. The second column shows the error of the classifier. The greater value of the error indicates a stronger attack. The loss of classifier is reported in the first column of Table 1. The third column shows the attack success rate. It's highest for our proposed approach, other than corner search but its ℓ_2-norm is much higher than all other approaches. In the case of Imagenet the FGSM has a higher success rate but at the cost of a higher ℓ_2-norm than all other approaches.

2) Targeted Attack

The results for the proposed targeted attacks are reported in Table 2. The mean and median value of 0.1 is reported for MNIST and 0.17, 0.002 for Imagenet. These are the results of the Average case where each image is attacked by different classes of images and in the end, the average result of all attacks is reported. The second column shows the loss. The lower the value of loss the stronger the attack is. Our approach has the lowest value of loss of any state-of-the-art. The values reported suggesting that our proposed attack performs better than the FGSM method and is as good as C&W. Although C&W is a very strong attack, it is computationally very expensive. We computed the run-time of an un-targeted C&W attack to be 4,303 seconds for MNIST, on a machine with an Intel(R) Core(TM) i7-7th generation CPU and 8GB of RAM. In contrast, the run-time for our proposed un-targeted attack is 245 seconds on the same machine. Hence, C&W attack is an order of magnitude slower than the presented method.

The third column in Table 2 records the attack success rate. Here an error with a low value implies a stronger targeted attack. In the case of MNIST, the attack success values are quite promising for FGSM and C&W. This is because the epsilon value is kept very low for FGSM. The C&W is the most effective targeted attack and reports the lowest(best) values for mean and median ℓ_2, but it needs a lot of iterations which makes them infeasible [7].

In a nutshell, the proposed method is competitive with state-of-the-art in terms of performance. The C&W outperforms in terms of ℓ_2-norm , but requires a lot of iterations. Moreover, it lacks performance in terms of loss and attack success. The detailed illustration is provided in Fig. 3. The feature map is used to create a perturbation vector and is then transformed into sparse representation as shown in Fig. 3 both for MNIST and Imagenet examples.

V. ABLATION STUDIES & ANALYSIS OF RESULTS

The critical analysis of results reported in the previous section is explained in this section with the help of ablation studies. Dictionary learning is the key to why we achieve promising results reported in the previous section. C&W provides a very strong state of the art targeted attack with minimum ℓ_2 distance but requires thousands of iterations making it infeasible. On the other hand, we achieved promising results by improving the efficiency of dictionary learning by training it on diverse and sampled feature maps. The results can be better explained by learning the effect of hyper-parameters of the dictionary. Hyper-parameters of the dictionary learning algorithm can be used to optimize its performance. The proposed tuned dictionary learning algorithm has two hyper-parameters: sparsity k, and dictionary size n, i.e. no. of components. They affect the performance in different ways. We compute the following proximity metric to compute the performance of the dictionary learning algorithm for different hyper-parameters.

Squared Error: is the Euclidean distance between the transformed image and the image associated with the target label. The lower value of the squared error means less difference between the transformed and original image and it helps us achieve adversarial images with a much smaller ℓ_2-norm. Figure 5. The experiments are conducted on all the test images of the MNIST dataset. The authors in [33] showed that increasing the sparsity, helps preserve more details but are less robust against attacks as well as increasing the no. of components also decreases the robustness of classifiers. We analyze the effect of sparsity k on the squared error in Fig. 4a. The difference between transformed and original image i.e. squared error is increased by increasing the sparsity. So sparsity can be used as a trade-off parameter here for targeted attacks.

We also studied the effect of the dictionary size i.e. the number of components as illustrated in Fig. 4b. We computed the values for squared error for $k = 1$ and $k = 3$ as the number of components of the dictionary are increased. It first decreases as the number of the components increase but starts increasing again for $k = 1$. It almost attains no further change in the squared error after the number of the components are increased till 255. When $k = 3$, the error attains stability earlier at the number of components 144 and further starts increasing after $n = 361$. Increasing the number of components improves the reconstruction and hence the accuracy on the clean images. It can be inferred from the

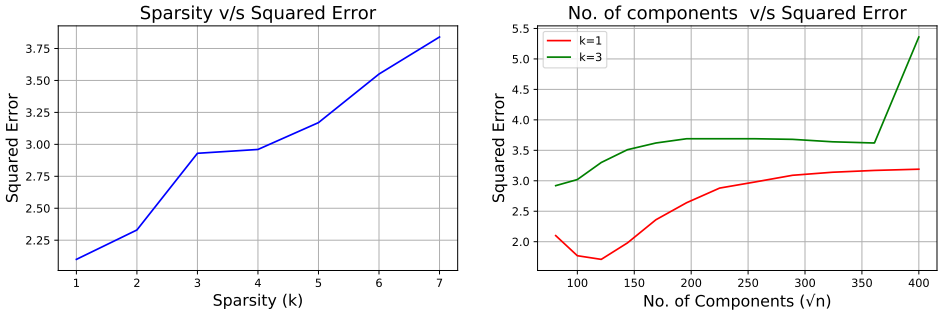

FIGURE 4. Left: The graph shows the effect of increasing sparsity k on Squared Error (Euclidean distance) between transformed and original targeted images. Right: The graph shows the effect of increasing dictionary size on squared error. The values are reported for sparsity, $k = 1$ and $k = 3$ for MNIST.

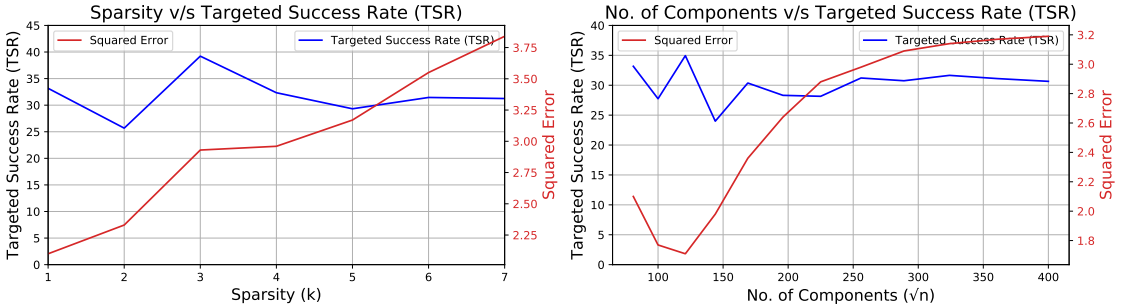

FIGURE 5. Left: The graph shows the effect of increasing sparsity k on the targeted success rate and squared error. Right: The effect of increasing dictionary size on targeted success rate and squared error with sparsity $k = 1$ for MNIST.

visual analysis that we get the desired result at a smaller size of the dictionary. This is in contrast to a regular trend in the literature because our work is not a reconstruction task where increasing the dictionary size increases accuracy on clean images. This way we save the computation cost, which increases as dictionary size increases in other regular tasks.

Next, we study the effect of these parameters on our attack strategy. The experiments in ablation studies show that squared error is also highly correlated with the targeted success rate (TSR) explained earlier in the metrics section. The effect of sparsity on TSR is illustrated in Fig. 5a. The TSR oscillates in the beginning and shows stability later after $k = 4$ as shown in the graph. The TSR attains the highest value for $k = 3$ and starts decreasing afterward. We see that at $k = 3$, more information is retained as compared to $k = 1$. Therefore, more noise is reconstructed as we increase the sparsity but increasing sparsity further increases the squared error i.e. the distance between original and transformed so it negatively affects the targeted attack. The peak at $k = 3$ in Fig. 5a, shows that we achieve maximum value for the targeted attack. After $k = 3$ the noise starts reconstructing as we are learning the dictionary of feature maps. When the noise gets reconstructed it does have a more strong attack on classifier but at the same time, the targeted attack also suffers from the reconstruction of more and more noise. That is why the squared error also increases showing that less important information is preserved and the ℓ_2-norm also increases. Therefore, $k = 3$ serves as a sweet spot in this case. The above details emphasize to trade off the sparsity to make an effective attack in terms of TSR as well as squared error. The experiments show $k = 3$ is the optimal value for this case. Therefore, we used both $k = 1$ and $k = 3$ to study the effect of the no. of the components on TSR explained in Fig. 5b and Fig. 6.

The same behavior can be seen for the number of the components. TSR increases at first for an increase in the dictionary size, but subsequently decreases. For $k = 3$ as the sparsity level is already high so TSR is the highest even for $n = 81$. For better understanding, we have again plotted squared error with TSR in Figures 6 and 7. It can be seen that the highest TSR is reported for lowest squared error which is the reason we achieve the targeted and un-targeted misclassifications with a very low ℓ_2-norm as reported earlier in the results section. In conclusion, we need to have a smaller squared error but a very small value will not retain enough information for the targeted attack. A very high value of squared error will again result in a higher ℓ_2-norm and low TSR. This behavior of reconstructing noise as we increase sparsity and the no. of the components is attributed to the fact that we are in fact, learning the dictionary of perturbation vectors. We also conducted experiments to check the effect of choosing feature maps from other layers of CNN. The results are illustrated in Table3

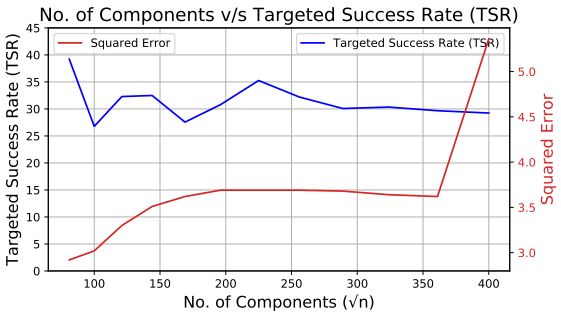

FIGURE 6. The effect of increasing dictionary size on targeted success rate and squared error with sparsity $k = 3$ for MNIST.

TABLE 3. The classifier's loss , attack success, mean and median ℓ_2-norm when feature maps are from other layer of the DNN.

MNIST				
CNN Layer	**Loss on Test**	**Attack Succ.**	**Mean ℓ_2**	**Median ℓ_2**
Layer 1	9.26	86%	0.1	0.1
Layer 2	2.64	44%	0.09	0.09
Layer 3	5.45	77%	0.1	0.1
Layer 4	3.10	67%	0.2	0.2

VI. DEFENSE EVALUATION AGAINST THE PROPOSED ATTACK

Devising defense strategies against adversarial attacks is an equally active area of research just like adversarial attacks. Goodfellow et al. [5] proposed the method of adversarial training in which the model is trained using adversarial images. In order, to evaluate the strength of our proposed attack we tested it against various defense methods. We used three different defense strategies to measure the effectiveness of our attack. **Spatial Smoothing [37]** is a technique used in image processing to reduce noise in the data. The authors in [37] applied the local smoothing method as a defense against attacks. Local Smoothing smooths each pixel by using neighboring pixels. **Feature Squeezing [37]** is used to reduce the bit depth of images. Images are normally represented using color bit depths which is a major cause of irrelevant features. In this paper, the authors tested the hypothesis that reducing bit depth can reduce the effect of adversarial attacks without affecting classifiers' accuracy. The method is applied to each pixel. **JPEG Compression [38]** is also used as an effective defense technique. Its strength lies in its ability to eliminate high-frequency signal components. These are removed inside the square blocks of a particular image.

The training data as well as adversarial data are transformed using defense methods and are then evaluated on the same model architecture as employed in Section IV. The data is trained on transformed training data for 30 epochs for MNIST. In the case of Imagenet, only adversarial data is transformed due to computational complexity and the use of a pre-trained model. The results show that our attack is not defended by any of these defense methods. The fooling ratio of the classifier, when fed with sparse adversarial perturbations, is recorded in the second column of Table 4. The next columns show the fooling ratio after applying different defense strategies. It can be seen from Table **??** that our proposed attack has a success rate of 86%. The next columns show the fooling ratios after applying defense methods to the MNIST dataset.

TABLE 4. The fooling ratio of adversarial attack, Spatial Smoothing (SS) defense, Feature Squeezing (FS) defense, and JPEG Compression (JC) for our proposed attack.

Dataset	Attack Succ.	SS	FS	JC
MNIST	86	77	82	88.5
Imagenet	53	52.51	52.23	53.8

For MNIST, the fooling ratio remained the same for spatial smoothing, and JPEG compression whereas, it increases in case of feature squeezing. This is because the basic idea behind defense methods in general and feature squeezing, in particular, is to compare the model's prediction on the original sample with the same model's prediction on the sample after squeezing [37]. Since our proposed attack has already used a smaller subspace and is minimum in terms of ℓ_2-norm therefore, feature squeezing didn't help. Moreover, the analysis in [37] shows that feature squeezing is not immune to adversarial adaptation and hurts the accuracy of legitimate images as well.

In the case of Imagenet again the defense methods failed to counter the effect of our proposed attack. It is reduced to 52.51% and 52.23% in the case of spatial smoothing and feature squeezing but is still not able to provide an effective defense. The emphasis of our approach has been on squeezing the noise magnitude. The important pixels are there but the sparse transformation of noise and lower value of ℓ_2-norm has made it almost difficult to detect the attack.

VII. CONCLUSION

We propose sparse adversarial image generation which obtains comparable results in terms of ℓ_2-norm. The feature map makes it possible to highlight the important pixels of the image to attack. We used a feature map to create our perturbation vector. This perturbation vector is then optimized using dictionary learning. The sparse adversarial noise is then added to the image by the one-shot method. The proposed attack is fast and minimum in terms of ℓ_2 distance between the input image and the adversarial image. The tables show that our results are comparable with state-of-the-art methods. There is room for improvement in terms of the fooling ratio of the attack. The comparable results are achieved using a small size of dictionary thus saving the computation cost. We motivate a new area for designing adversarial attacks not explored before. The researchers can explore this area to create more robust classifiers.

Further, we tested the strength of our proposed attack with different defense strategies. The results show that these defense methods are not able to defend neural networks from our proposed attack. Since this is a new direction to create adversarial examples. In the future, the proposed attack can be combined with existing gradient-based attacks and can be used in adversarial training to create more robust classifiers. We have also presented novel algorithms to learn tuned dictionary based on feature maps. These ideas to tune dictionaries can be extended to other machine learning problems solved by dictionary learning. This research is still needs improvement in terms of attack success rate, especially in the case of targeted attacks. Since this area was not explored yet, the future avenues hold strong. One stream of work can be conducted to improve the results in terms of attack success rate. The other is to test the transferability of these attacks to other models as well as other machine learning problems.

REFERENCES

[1] C. Szegedy, S. E. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," *CoRR*, vol. abs/1412.1441, 2014.

[2] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. D. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *2nd International Conference on Learning Representations, ICLR, Banff, AB, Canada, April 14-16*, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations, ICLR, Banff, AB, Canada, April 14-16*, 2014.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, May 7-9*, 2015.

[6] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[7] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4322–4330.

[8] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[9] Y. Shi, S. Wang, and Y. Han, "Curls & whey: Boosting black-box adversarial attacks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6519–6527.

[10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[11] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033–1040.

[12] K. Skretting and K. Engan, "Image compression using learned dictionaries by rls-dla and compared with k-svd," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 1517–1520.

[13] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent k-svd: Learning a discriminative dictionary for recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2651–2664, 2013.

[14] Y. Chen and J. Su, "Sparse embedded dictionary learning on face recognition," *Pattern Recognition*, vol. 64, pp. 51–59, 2017.

[15] 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.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[17] S. U. Din, N. Akhtar, S. Younis, F. Shafait, A. Mansoor, and M. Shafique, "Steganographic universal adversarial perturbations," *Pattern Recognition Letters*, pp. 146–152, 2020.

[18] Z. Yahya, M. Hassan, S. Younis, and M. Shafique, "Probabilistic analysis of targeted attacks using transform-domain adversarial examples," *IEEE Access*, vol. 8, pp. 33 855–33 869, 2020.

[19] C. Zhang, P. Benz, T. Imtiaz, and I. S. Kweon, "Understanding adversarial examples from the mutual influence of images and perturbations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 521–14 530.

[20] M. Li, C. Deng, T. Li, J. Yan, X. Gao, and H. Huang, "Towards transferable targeted attack," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 641–649.

[21] N. Inkawhich, W. Wen, H. H. Li, and Y. Chen, "Feature space perturbations yield more transferable adversarial examples," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7066–7074.

[22] M. A. Jalwana, N. Akhtar, M. Bennamoun, and A. Mian, "Attack to explain deep representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9543–9552.

[23] Y. Shi, Y. Han, Q. Zhang, and X. Kuang, "Adaptive iterative attack towards explainable adversarial robustness," *Pattern Recognition*, p. 107309, 2020.

[24] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

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

[26] F. Croce and M. Hein, "Sparse and imperceivable adversarial attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4724–4732.

[27] X. Dong, D. Chen, J. Bao, C. Qin, L. Yuan, W. Zhang, N. Yu, and D. Chen, "Greedyfool: Distortion-aware sparse adversarial attack," *arXiv preprint arXiv:2010.13773*, 2020.

[28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[29] Z. He, W. Wang, J. Dong, and T. Tan, "Transferable sparse adversarial attack," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 963–14 972.

[30] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[31] J. Mitro, D. Bridge, and S. Prestwich, "Denoising dictionary learning against adversarial perturbations," *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 364–371, 2018.

[32] S. Gopalakrishnan, Z. Marzi, U. Madhow, and R. Pedarsani, "Combating adversarial attacks using sparse representations," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, 2018.

[33] S. Moosavi-Dezfooli, A. Shrivastava, and O. Tuzel, "Divide, denoise, and defend against adversarial attacks," *CoRR*, vol. abs/1802.06806, 2018.

[34] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," in *4th International Conference on Learning Representations, ICLR, San Juan, Puerto Rico, May 2-4*, 2016.

[35] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[36] M.-I. Nicolae, M. Sinn, M. N. Tran, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy *et al.*, "Adversarial robustness toolbox v0. 4.0," *arXiv preprint arXiv:1807.01069*, 2018.

[37] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*, 2018.

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

[39] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering adversarial images using input transformations," in *6th International Conference*

on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.

[40] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 501–509.

[41] G. Cohen, G. Sapiro, and R. Giryes, "Detecting adversarial samples using influence functions and nearest neighbors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 453–14 462.

[42] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

MAHAM JAHANGIR is a PhD Scholar working under supervision of Professor Dr. Faisal Shafait at School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan. She received her MS Degree from the Military College of Signal (MCS), NUST. Her research interests include machine learning with special interests in deep neural networks and adversarial attacks.

FAISAL SHAFAIT , currently working as a Professor at School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan. Besides, Previously, He was an Adjunct Senior Lecturer at the School of Computer Science and Software Engineering at The University of Western Australia in Perth, Australia. He was also a Senior Researcher at the German Research Center for Artificial Intelligence (DFKI) as well as an Adjunct Lecturer at Kaiserslautern University of Technology (TUKL), Germany. He received his PhD with the highest distinction in computer engineering from TUKL in 2008. His research interests include machine learning and pattern recognition with a special emphasis on applications in document image analysis.