Mimic and Fool: A Task-Agnostic Adversarial Attack

Akshay Chaturvedi and Utpal Garain, *Member, IEEE*

***Abstract*—At present, adversarial attacks are designed in a task-specific fashion. However, for downstream computer vision tasks such as image captioning and image segmentation, the current deep-learning systems use an image classifier such as VGG16, ResNet50, and Inception-v3 as a feature extractor. Keeping this in mind, we propose Mimic and Fool (MaF), a task-agnostic adversarial attack. Given a feature extractor, the proposed attack finds an adversarial image, which can mimic the image feature of the original image. This ensures that the two images give the same (or similar) output regardless of the task. We randomly select 1000 MSCOCO validation images for experimentation. We perform experiments on two image captioning models, Show and Tell, Show Attend and Tell, and one visual question answering (VQA) model, namely, end-to-end neural module network (N2NMN). The proposed attack achieves a success rate of 74.0%, 81.0%, and 87.1% for Show and Tell, Show Attend and Tell, and N2NMN, respectively. We also propose a slight modification to our attack to generate natural-looking adversarial images. In addition, we also show the applicability of the proposed attack for invertible architecture. Since MaF only requires information about the feature extractor of the model, it can be considered as a gray-box attack.**

***Index Terms*—Adversarial attack, deep learning, task agnostic method, vision and language systems.**

I. Introduction

Adversarial attacks have shed light on the vulnerability of several state-of-the-art deep-learning systems across varied tasks, such as image classification, object detection, and image segmentation. [1]–[4]. Recently, adversarial attacks were also proposed for multimodal tasks involving vision and language such as image captioning and visual question answering (VQA) [5], [6]. Usually, these attacks fall under two categories: white box and black box. In the white-box attack, the adversary has complete information about the model and its parameters, whereas in the black-box attack, the adversary has no information about the model that it wants to attack. Black-box attacks [7] are possible due to the transferability phenomenon of adversarial examples. Liu *et al.* [8] showed that the adversarial examples designed for one image classification model can be transferred successfully to other classification models as well. Similarly, Xu *et al.* [6] showed the transferability of adversarial images between two state-of-the-art VQA models. Very recently, Shi *et al.* [9] improved the black-box attack performance for image classification by allowing for more diverse search trajectories and squeezing redundant noise. However, the present-day adversarial attacks are task-specific in nature since a task-specific adversarial loss function is optimized to generate adversarial examples.

On the other hand, the current deep-learning systems use output from intermediate layers of convolutional neural network (CNN)-based image classification models (e.g., ResNet50 [10], VGG16 [11], and Inception-v3 [12]) as a feature for the input image. The rationale behind this approach is that the discriminative features learned by these classifiers are useful for other vision tasks as well. Hence, it is more beneficial to use these features instead of learning them from scratch. As a result, the aforementioned image classifiers function as feature extractors. Some deep-learning systems also fine-tune the parameters of the feature extractors during training to make the image feature more suitable for the task in hand. However, fine-tuning is usually done if a large amount of training data is available. Although using deep CNN-based image features give a significant advantage to the present-day models, they have their own set of drawbacks. CNN-based feature extractors are known to be noninvertible [13], [14]. Mahendran and Vedaldi [14] showed that AlexNet [15] maps multiple images to the same 1000-D logits. These images are thus indistinguishable from the viewpoint of the last fully connected layer of AlexNet.

A. Motivation: Agnosticism in Adversarial Attacks

The main goal of this brief is to introduce the notion of a task-agnostic attack. If such an attack were possible, it will shed light on the common weakness shared by different vision systems across various tasks. So far, adversarial training has been the most popular approach for building robust image classifiers. However, adversarial training is computationally expensive and, more importantly, ill-defined for downstream tasks such as image captioning. In such a scenario, mitigating the common weakness can make the task of building robust end-to-end systems tractable.

In this brief, we propose Mimic and Fool (MaF), a task-agnostic adversarial attack, which exploits the noninvertibility of CNN-based feature extractors to attack the downstream model. Given a model and its feature extractor, the proposed attack is based on the simple hypothesis that if two images are indistinguishable for the feature extractor, then they will be indistinguishable for the model as well. In other words, attacking the feature extractor by finding two indistinguishable images is equivalent to hacking the eyes of the model. As an example, consider an encoder–decoder architecture such as Show and Tell [16]. If we can successfully find two images that are mapped to the same feature by the encoder, then the two images will generate the same (or similar) caption regardless of the decoder architecture. Thus, to attack any model, attacking its feature extractor suffices. Based on this insight, MaF finds an adversarial image, which can mimic the feature of the original image, thereby fooling the model. Fig. 1 shows the examples of MaF on two captioning models: Show and Tell [16], Show Attend and Tell [17], and one VQA model: end-to-end neural module network (N2NMN) [18]. It is crucial to note that the goal of MaF differs from traditional adversarial attacks [2], [3], [5], [6]. In traditional adversarial attacks, a small amount of noise is added to the image in order to fool the model to generate a different output, whereas in MaF, the goal is to generate an adversarial image that can fool the model to predict the same output as the original image. As we can see from Fig. 1, the adversarial images obtained via MaF are noisy images. Such images, although noisy, pose a security risk for real-world systems. This is in line with adversarial attacks on object detectors where a large amount of noise is added [1], [19]. In order to generate natural-looking adversarial images, we also propose a modified version of our attack, namely One Image Many

Manuscript received May 20, 2019; revised November 20, 2019 and March 3, 2020; accepted March 28, 2020. (*Corresponding author: Akshay Chaturvedi.*)

The authors are with the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata 700108, India (e-mail: akshay91.isi@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2984972

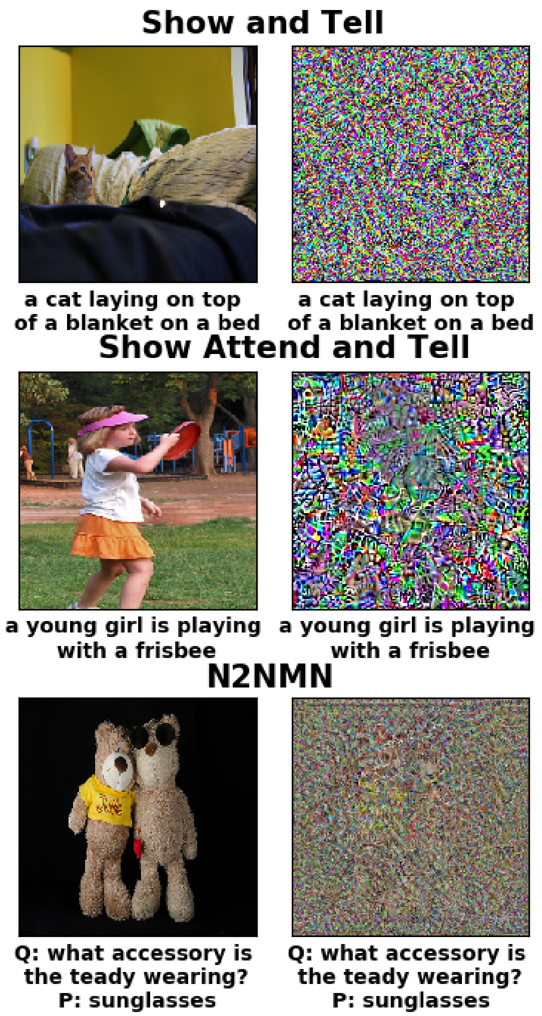

Show and Tell
a cat laying on top of a blanket on a bed
a cat laying on top of a blanket on a bed
Show Attend and Tell
a young girl is playing with a frisbee
a young girl is playing with a frisbee
N2NMN
Q: what accessory is the teady wearing? P: sunglasses
Q: what accessory is the teady wearing? P: sunglasses

Fig. 1. Examples of MaF. The first two rows show the original and adversarial images along with the predicted captions by Show and Tell and Show Attend and Tell, respectively. The last row shows the original and adversarial images for N2NMN (Q and P denote the question and the predicted answer, respectively).

Outputs (OIMO). In OIMO, we start with a fixed natural image and restrict the amount of noise that can be added to the image.

Since MaF only requires the fine-tuned weights of the feature extractor to attack the model, it can be thought of as a gray-box attack. In fact, if a model does not fine-tune its feature extractor, MaF can function as a black-box attack. This is because the number of possible feature extractors is limited. Hence, an adversary can generate an adversarial image per feature extractor knowing that one of these images is bound to fool the model. Furthermore, MaF is extremely fast and requires less computing resources since only the feature extractor needs to be loaded in the memory instead of the model.

We perform experiments on two tasks: image captioning and VQA. We randomly choose 1000 MSCOCO [20] validation images and study the proposed attack on three models: Show and Tell, Show Attend and Tell, and N2NMN. We get 5208 image-question pairs from VQA v2.0 data set [21] for the 1000 selected images. We choose these three models since they use different feature extractors. Show and Tell uses fully connected features from Inception-v3, Show Attend and Tell uses convolutional layer features from VGG16, and N2NMN uses features from a residual network [10]. Thus, the three feature extractors vary from shallow to very deep, which helps us to validate our proposed attack for different types of feature extractors. We consider our attack successful if the model gives the same output for the original and adversarial images.

B. Contributions of This Brief

The contributions of this brief are as follows.

1) We introduce the notion of a task-agnostic attack. The proposed task-agnostic attack, MaF, achieves high success rates for Show and Tell, Show Attend and Tell, and N2NMN. This validates our hypothesis that attacking the feature extractor suffices and also shows that the proposed attack works for different feature extractors. For image captioning models, we also compute the BLEU [22] and METEOR [23] score for the failure cases to show that even though the original and adversarial captions do not match exactly for these cases, they are very similar to each other.
2) Even for OIMO, the proposed attack achieves a decent success rate. This shows that, by adding minimal noise to the fixed image, it is possible to find an adversarial image that can mimic the image feature of any arbitrary image. This result is intriguing as it suggests that the feature extractors are very chaotic in nature.
3) Since MaF is task-agnostic, we need to run it for every image instead of every image-question pair to attack a VQA model such as N2NMN. This is a huge advantage in terms of time saved for the adversary. The same will hold true for any future tasks which take multiple modalities as input with image being one of the modalities.
4) At first glance, it seems that an invertible feature extractor will be resistant to the proposed attack. However, we show that the proposed attack also works for invertible architecture [24]. This shows that such architectures, despite being invertible, assign similar features to dissimilar images. Hence, invertibility is not a sufficient condition to safeguard the models against the proposed attack.

II. Method

A. Proposed Attack

In this section, we describe the proposed attack, MaF and OIMO that can generate natural-looking adversarial images. Since both the attacks are task-agnostic, we describe the attack in terms of the feature extractor instead of the model.

1) Mimic and Fool: Let $f : \mathbb{R}^{m\times n\times 3} \longrightarrow \mathbb{R}^d$ denote the feature extractor of the model. Hence, d will be $14 \times 14 \times 1024$ if we extract conv4 features from ResNet101 and d will be 2048 if we use output of average pooling layer of Inception-v3 as image feature.

Let $I_{\text{org}} \in [0, 255]^{m\times n\times 3}$ denote the original image. Given I_{org} and a feature extractor f, our goal is to find an adversarial image $I_{\text{adv}} \in [0, 255]^{m\times n\times 3}$, which can mimic the image features of I_{org}. We model this task as a simple optimization problem given by

$$\min_{I} \frac{\| f(\text{trunc}(I)) - f(I_{\text{org}})\|_2^2}{d} \tag{1}$$

where $\| \cdot \|_2$ denotes ℓ_2-norm and trunc is truncating function, which ensures that the intensity values lie in the range $[0, 255]$. Although $I = I_{\text{org}}$ is a solution to the abovementioned optimization problem, it is highly unlikely that the algorithm will converge to this solution. This is because CNNs discard significant amount of spatial information as we go from lower to higher layers. Mahendran and Vedaldi [14] showed that the amount of invariance increases from lower to higher layer of AlexNet and regularizers, such as total variation (TV), are needed to reconstruct the original image from higher layer features of AlexNet. We start with a zero image and run the proposed attack for $\max_{\text{iter}}$ iterations and return the final truncated image $\text{trunc}(I)$ as I_{adv}.

Some feature extractors, such as Inception-v3, require the intensity values of the input image to be in the range $[-1, 1]$. In such a case, let $I'_{\text{org}} \in [-1, 1]^{m \times n \times 3}$ be the scaled original image, that is

$$I'_{\text{org}} = 2(I_{\text{org}}/255) - 1. \tag{2}$$

For this case, we modify the optimization problem defined in 1 as follows:

$$\min_{I} \frac{\left\| f(\tanh(I)) - f\left(I'_{\text{org}}\right) \right\|_2^2}{d} \tag{3}$$

where tanh ensures that the input to feature extractor lies within the required range. We run the attack for $\max_{\text{iter}}$ iterations and rescale the final image $tanh(I)$ to get I_{adv}, that is

$$I_{\text{adv}} = 255 \left(\frac{\tanh(I) + 1}{2} \right). \tag{4}$$

2) One Image Many Outputs: In OIMO, we start with an image $I_{\text{start}} \in [0, 255]^{m \times n \times 3}$ instead of starting with zero image. The image I_{start} is kept fixed throughout the experiment. In OIMO, our goal is to modify I_{start} so as to mimic the feature of I_{org}. Equation 1 is modified as follows:

$$\min_{\delta} \frac{\left\| f(\text{trunc}(I_{\text{start}} + \delta)) - f(I_{\text{org}}) \right\|_2^2}{d}. \tag{5}$$

Similar to Chen *et al.* [5], we modify 3 as follows:

$$\min_{\delta} \frac{\left\| f\left(\tanh\left(I''_{\text{start}} + \delta\right)\right) - f\left(I'_{\text{org}}\right) \right\|_2^2}{d} \tag{6}$$

where $I''_{\text{start}} = \ \text{arctanh}(\lambda I'_{\text{start}})$, $I'_{\text{start}} \in [-1, 1]^{m \times n \times 3}$ is the scaled starting image, λ is set to 0.9999 to ensure invertibility of tanh, and $\delta \in \mathbb{R}^{m \times n \times 3}$ is the learnable parameter. For this attack, we reduce the value of $\max_{\text{iter}}$ and initial learning rate to ensure that I_{adv} looks very similar to I_{start}.

Similar to MaF, after running the attack for $\max_{\text{iter}}$ iterations, I_{adv} for 5 is $\text{trunc}(I_{\text{start}} + \delta)$. For 6, I_{adv} is given by the following equation:

$$I_{\text{adv}} = 255 \left(\frac{\tanh\left(I''_{\text{start}} + \delta\right) + 1}{2} \right). \tag{7}$$

We name the proposed attack OIMO since all the adversarial images look very similar to I_{start}.

B. Implementation Details

As stated earlier, we study the proposed attack for two image captioning models: Show and Tell, Show Attend and Tell, and one VQA model, namely, N2NMN. We train the N2NMN model on the VQA v2.0 data set for 95k iterations with expert policy followed by 65k iterations in policy search after cloning stage using the original source code.[1] The trained N2NMN has 61.72% accuracy on VQAv2 test-dev set. For Show and Tell and Show Attend and Tell, we use already available trained models.[2,3]

Show and Tell uses 2048-D feature from Inception-v3, Show Attend and Tell uses $14 \times 14 \times 512$ feature map from VGG16, and N2NMN uses output of *res*5*c* layer from ResNet-152 as image feature. The input images are of size $299 \times 299 \times 3$, $224 \times 224 \times 3$, $448 \times 448 \times 3$ for Inception-v3, VGG16, and ResNet-152, respectively. The trained Show and Tell and Show Attend and Tell fine-tune their respective feature extractors, whereas N2NMN does not use fine-tuning.

For MaF, we set $\max_{\text{iter}}$ to 1000, 1000, and 2000 for Inception-v3, VGG16, and ResNet-152, respectively. The initial learning rate is set to 0.025, 0.025, and 0.0125 for Inception-v3, VGG16, and ResNet-152, respectively. For OIMO, we set $\max_{\text{iter}}$ to 300, 500, and 500 and set the initial learning rate to 0.0125, 0.0125, and 0.00625 for Inception-v3, VGG16, and ResNet-152, respectively. We use Adam [25] as the optimizer and Keras [26] for implementing the proposed attacks. All experiments are done on a single 11 GB GeForce GTX 1080 Ti GPU. The code for MaF is publicly available.[4]

III. Results

For studying the two proposed attacks, 1000 MSCOCO validation images are randomly selected. For the 1000 selected images, there are 5208 image-question pairs in the VQA v2.0 data set. For VQA, we discard those image-question pairs where the VQA model predicts the same answer for I_{start} and I_{org} (for MaF, I_{start} is zero image). This is done to ensure that the VQA model predicts the same answer for I_{start} and I_{org} due to adversarial noise rather than language bias. The proposed attack is considered to be successful if the model gives the same output for the original and the adversarial image. Hence, for image captioning, the two captions need to be exactly the same for the attack to be successful. In this section, we analyze the behavior of the two proposed attacks on the three models: N2NMN, Show and Tell, and Show Attend and Tell. We also study the effectiveness of the proposed method for an invertible architecture.

A. Results for Mimic and Fool

Table I shows the success rate of MaF for the three models. Out of 5208 image-question pairs, N2NMN predicts the same answer for I_{org} and zero image for 1707 pairs. Out of the remaining 3501 pairs, MaF is successful for 3049 image-question pairs. This yields a success rate of 87.1%. The high success rate shows that it is possible to mimic features extracted from a very deep network such as ResNet-152 as well. Since MaF is task-agnostic, we need to run the proposed attack at the image level instead of the image-question pair level. This is a huge advantage since it results in a drastic reduction in time. The advantage will be even more pronounced for any future tasks which have multiple modalities as input with image (or video) being one of the modalities. Fig. 2 shows the predicted answer by N2NMN for different image-question pairs. From Fig. 2, we can see that a single adversarial image suffices for three image-question pairs.

As we can see from Table I, MaF is very fast. The attack only takes around 25 s for generating adversarial images for Show and Tell. The time taken for Show, Attend, and Tell is even less since VGG16 is a shallower network. The proposed attack achieves the success rate of 74.0% and 81.0% for Show and Tell and Show Attend and Tell, respectively. This is especially encouraging result since generating exactly the same caption for an adversarial image is a very challenging task. This is because, as observed by Chen *et al.* [5], the number of possible captions is infinite, which makes a captioning system harder to attack than an image classifier. Our results show that in order to generate the same caption, it suffices to attack just the encoder of the captioning model. This validates our initial hypothesis that in order to attack any model, attacking its feature extractor suffices. For the unsuccessful cases, the predicted captions for original and adversarial images are very similar. We also calculate the BLEU

[1] https://github.com/ronghanghu/n2nmn

[2] https://github.com/KranthiGV/Pretrained-Show-and-Tell-model

[3] https://github.com/DeepRNN/image_captioning

[4] https://github.com/akshay107/mimic-and-fool

TABLE I
SUCCESS RATE OF MAF

Task	Model	Feature Extractor	Success Rate	Average Time for 1000 iterations
Image Captioning	Show and Tell	Inception-v3	74.0 %	25.35 sec
	Show Attend and Tell	VGG16	81.0 %	15.56 sec
VQA	N2NMN	ResNet-152	87.1 %	72.98 sec

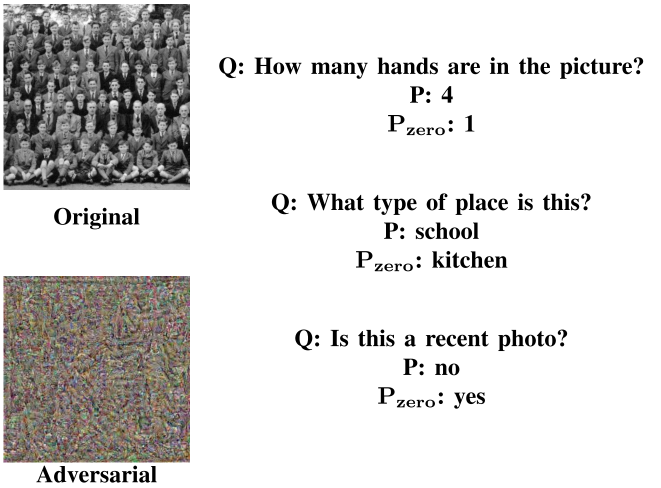

Original
Adversarial
Q: How many hands are in the picture?
P: 4
P_{zero}: 1
Q: What type of place is this?
P: school
P_{zero}: kitchen
Q: Is this a recent photo?
P: no
P_{zero}: yes

Fig. 2. Example of MaF for N2NMN. Single adversarial image suffices for three image-question pairs. Q and P denote the question and the predicted answer, respectively. P_{zero} denotes the predicted answer for zero image.

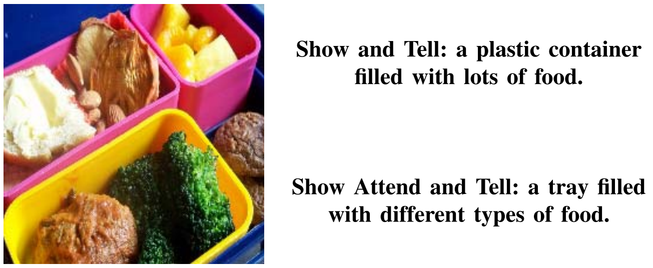

Show and Tell: a plastic container filled with lots of food.
Show Attend and Tell: a tray filled with different types of food.

Fig. 3. I_{start} for OIMO and the predicted captions.

and METEOR score, using the pipeline provided by Sharma *et al.* [27], for unsuccessful adversarial cases, as shown in Table II. We use the predicted caption for the original image as a reference while calculating these metrics.

B. Results for One Image Many Outputs

The main idea behind OIMO is to generate natural-looking adversarial images. We randomly choose an image from the MSCOCO training set as the starting image. Fig. 3 shows the starting image (I_{start}) for OIMO along with the predicted captions of Show and Tell and Show Attend and Tell. We use the same I_{start} for N2NMN. Similar to MaF, we discard 1713 image-question pairs for which N2NMN predicts the same answer for I_{org} and I_{start}.

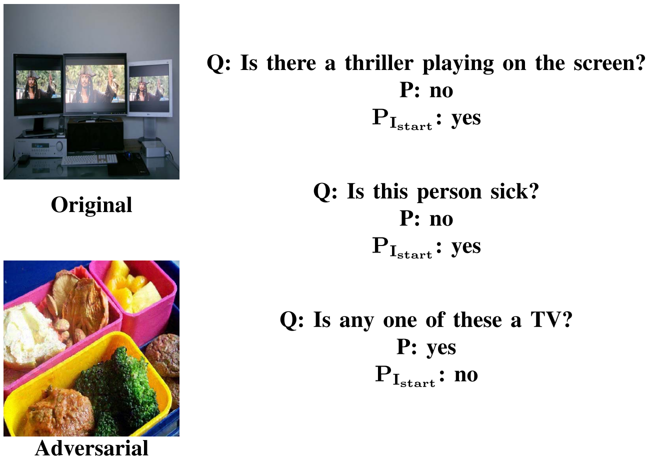


Fig. 4. Example of OIMO for N2NMN. Single adversarial image suffices for three image-question pairs. Q and P denote the question and the predicted answer, respectively. $P_{I_{\text{start}}}$ denotes the predicted answer for I_{start}.

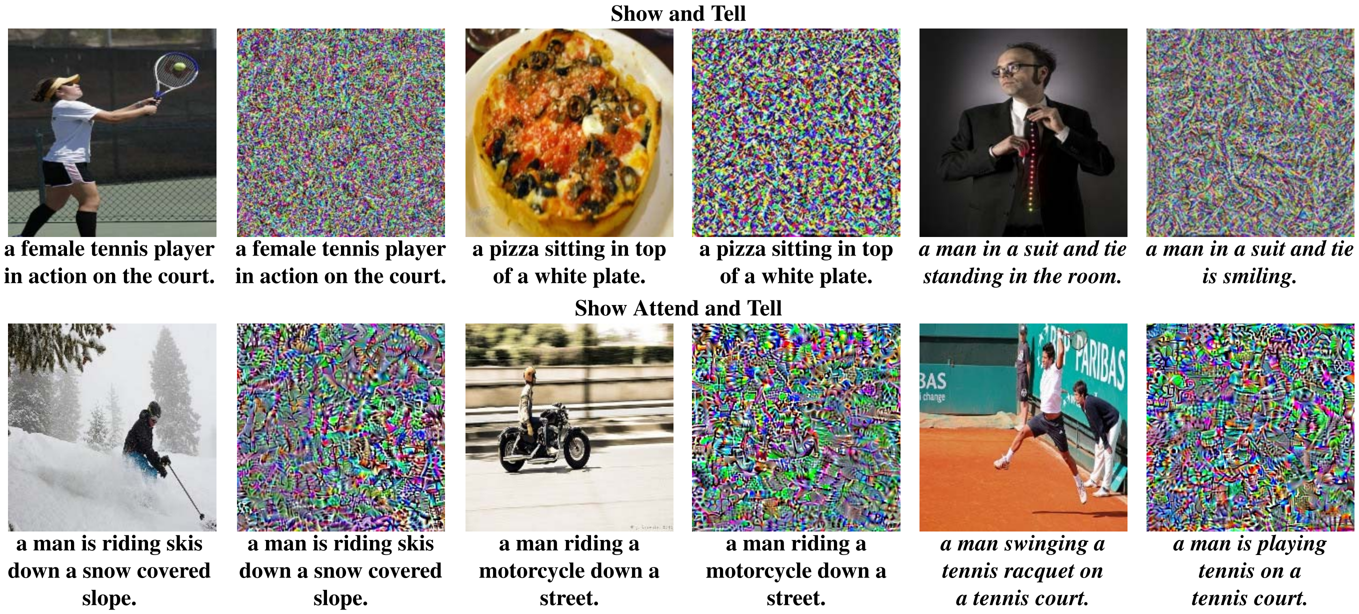
In OIMO, we reduce the value of $\max_{\text{iter}}$ and the initial learning rate to ensure that the adversarial image I_{adv} looks very similar to I_{start}. Reduction in $\max_{\text{iter}}$ results in even faster running time than MaF. Table III shows the success rate of OIMO for Show and Tell, Show Attend and Tell, and N2NMN. As we can see from Tables I and III, the success rate reduces for OIMO in comparison with MaF. This is intuitive since in OIMO, the reduced value of $\max_{\text{iter}}$ and initial learning rate allows for less adversarial noise. Fig. 4 shows an example of OIMO for N2NMN. Similar to MaF, a single adversarial image suffices for multiple image-question pairs.

From Table III, we can see that OIMO takes under 8 s per image for both the captioning models. Considering this reduction and the fact that the attack is successful only when there is an exact match of captions, the success rate of OIMO is impressive. Similar to MaF, we find that for the unsuccessful cases of OIMO, the captions predicted by the model for the adversarial and original images are very similar to each other. Table II shows the BLEU and METEOR score for the unsuccessful cases of OIMO. This result shows that even when I_{adv} is very similar to I_{start}, it can mimic features of an arbitrary image. This shows that CNN-based feature extractors are chaotic in nature.

Fig. 5 shows two successful and one unsuccessful examples of MaF for Show and Tell and Show Attend and Tell. As we can see from Fig. 5 that for the unsuccessful cases, the predicted captions for the original and adversarial images have a large amount of overlap. Fig. 6 shows two successful and one unsuccessful examples (shown in italics) of OIMO for Show and Tell and Show Attend and Tell. For the adversarial images in Fig. 6, ST and SAT denote Show and Tell and Show Attend and Tell, respectively. As we can see from Fig. 6, all the six adversarial images are very similar to the starting image, I_{start}. Also, for the unsuccessful cases, the original and adversarial captions have a large amount of overlap and are semantically similar. In Fig. 6, we see that for Show and Tell, the captions predicted by Show Attend and Tell for the three adversarial images are the same. Similarly, for Show Attend and Tell, although the captions predicted by Show and Tell are different, they are semantically similar. Moreover, for both the captioning models, the predicted captions by the other captioning model are relevant captions for the starting image, I_{start}. In fact, we find that when the 1000 adversarial images for Show And Tell are given as input to Show Attend and Tell, there are only 15 unique captions. All these 15 captions are relevant captions for I_{start}. Similarly, when the 1000 adversarial images for Show Attend

TABLE II
BLEU AND METEOR SCORES FOR UNSUCCESSFUL CASES. OIMO REFERS TO ONE IMAGE MANY OUTPUTS

Model	Attack	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Show and Tell	Show-and-Fool [5]	0.560	0.394	0.266	0.205	0.301
	Mimic and Fool	0.597	0.464	0.348	0.264	0.320
	OIMO	0.593	0.459	0.350	0.270	0.322
Show Attend and Tell	EM [28]	0.765	0.650	0.529	0.423	0.425
	SSVM [28]	0.635	0.501	0.409	0.300	0.337
	Mimic and Fool	0.639	0.530	0.421	0.333	0.368
	OIMO	0.594	0.468	0.359	0.284	0.336



Show and Tell

a female tennis player in action on the court.

a female tennis player in action on the court.

a pizza sitting in top of a white plate.

a pizza sitting in top of a white plate.

a man in a suit and tie standing in the room.

a man in a suit and tie is smiling.

Show Attend and Tell

a man is riding skis down a snow covered slope.

a man is riding skis down a snow covered slope.

a man riding a motorcycle down a street.

a man riding a motorcycle down a street.

a man swinging a tennis racquet on a tennis court.

a man is playing tennis on a tennis court.

Fig. 5. Examples of MaF. For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial images along with the predicted captions. Unsuccessful cases are shown in italics.

TABLE III
SUCCESS RATE OF OIMO

Model	Success Rate	Time (in sec.)
Show and Tell	56.9 %	7.61
Show Attend and Tell	50.3 %	7.78
N2NMN	72.8 %	36.50

and Tell are given as input to Show and Tell, there are only 82 unique captions, most of which are relevant to I_{start}. We find that Show and Tell generates irrelevant captions for I_{start} only for 32 out of 1000 adversarial images. Since the two captioning models use different feature extractors, this result shows that the proposed attack is very dependent on the feature extractor. In other words, ensuring that the two images are indistinguishable for one feature extractor does not ensure that they will be indistinguishable for another feature extractor. More examples of the two proposed attacks are provided in the Supplementary Material.[5]

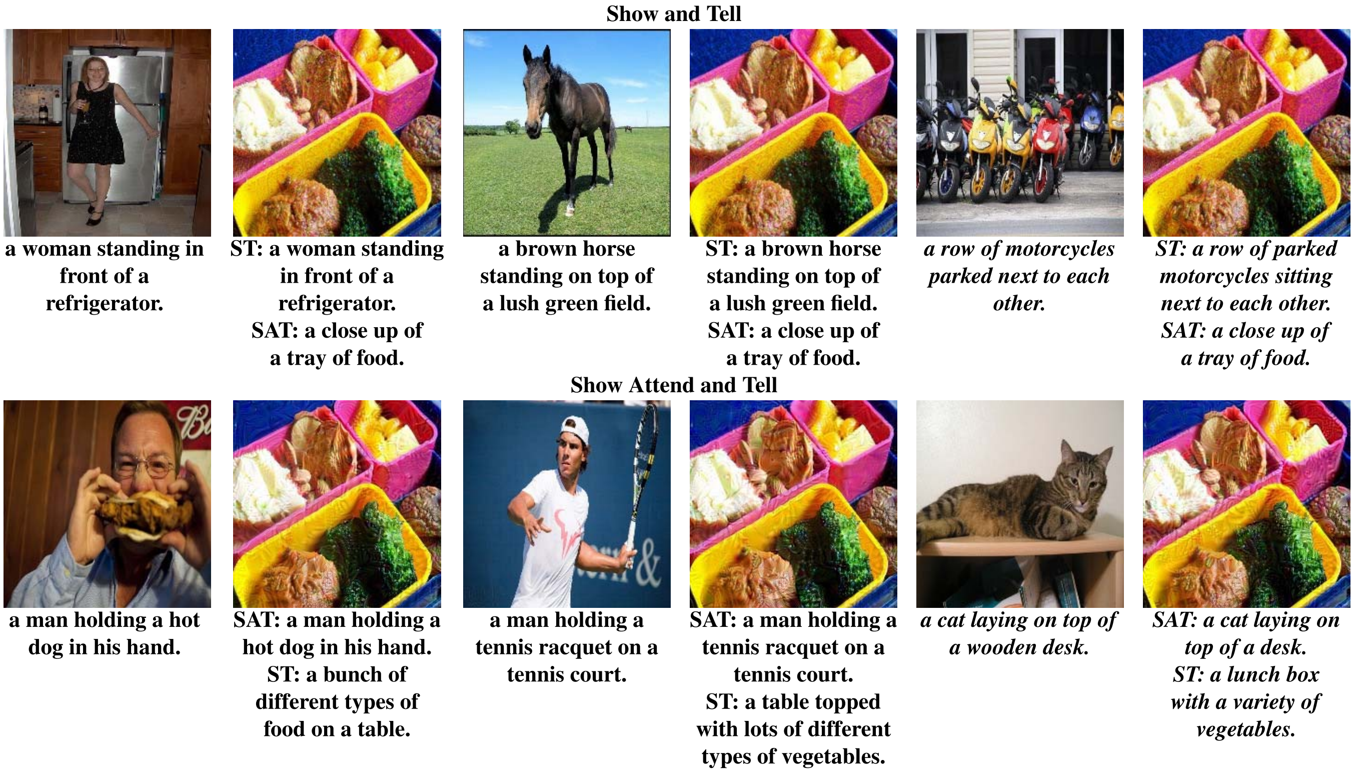
C. Comparison With Task-Specific Attack

In this section, we compare our proposed attack, OIMO, with other task-specific attacks. For Show and Tell, we use Show-and-Fool [5]. For Show Attend and Tell, we use EM and SSVM methods of Xu *et al.* [28]. For N2NMN, we use the VQA attack of Xu *et al.* [6]. For Show-and-Fool and EM and SSVM methods, we use the official implementation.[6,7] We implement the attack proposed by Xu *et al.* [6] using the default parameters mentioned in the brief. Similar to OIMO, we start with I_{start} and run the task-specific attacks in order to generate adversarial outputs. Table IV shows the success rate and time for different task-specific methods. Show-and-Fool achieves a success rate of 95.1% and takes 177.93 s per image. The EM and SSVM take less time for Show Attend and Tell but have lower success rates. In contrast, OIMO takes around 8 s per image for both the captioning models. For unsuccessful cases, such as OIMO, Show-and-Fool and EM and SSVM generate similar captions for original and adversarial images as evident from high BLEU and METEOR scores in Table II. We find that for the adversarial images generated by Show-and-Fool, Show Attend and Tell generates only 11 unique captions, all of which are relevant captions for I_{start}. Chen *et al.* [5] studied the transferability of Show-and-Fool between the captioning models; however, in their study, the two captioning models use the same feature extractor. Similarly, we obtain only three and five unique captions from Show and Tell for adversarial images of EM and SSVM, respectively. All these captions are relevant for I_{start}. Xu *et al.* [6] achieves a 100.0% success rate. The attack takes 8.77 s for each image-question pair. The factor n

[5] https://www.isical.ac.in/~utpal/resources.php

[6] https://github.com/IBM/Image-Captioning-Attack

[7] https://github.com/wubaoyuan/adversarial-attack-to-caption



Show and Tell

a woman standing in front of a refrigerator.

ST: a woman standing in front of a refrigerator.
SAT: a close up of a tray of food.

a brown horse standing on top of a lush green field.

ST: a brown horse standing on top of a lush green field.
SAT: a close up of a tray of food.

a row of motorcycles parked next to each other.

ST: a row of parked motorcycles sitting next to each other.
SAT: a close up of a tray of food.

Show Attend and Tell

a man holding a hot dog in his hand.

SAT: a man holding a hot dog in his hand.
ST: a bunch of different types of food on a table.

a man holding a tennis racquet on a tennis court.

SAT: a man holding a tennis racquet on a tennis court.
ST: a table topped with lots of different types of vegetables.

a cat laying on top of a wooden desk.

SAT: a cat laying on top of a desk.
ST: a lunch box with a variety of vegetables.

Fig. 6. Examples of OIMO. For both the captioning models, the figure shows two successful and one unsuccessful original and adversarial images along with the predicted captions. Unsuccessful cases are shown in italics. For adversarial images, ST and SAT denote Show and Tell and Show Attend and Tell, respectively.

TABLE IV

SUCCESS RATE AND TIME FOR TASK-SPECIFIC METHODS

Task	Model	Method	Success Rate	Time (in sec)
Image Captioning	Show and Tell	Show-and-Fool [5]	95.1 %	177.93
	Show Attend and Tell	EM [28]	77.1 %	20.69
		SSVM [28]	82.1 %	18.73
VQA	N2NMN	Xu et al. [6]	100.0 %	$8.77 \times n$

in the time for Xu *et al.* in Table IV signifies the average number of questions per image, which can be arbitrarily large.

D. OIMO for Invertible Architecture

Recently, Jacobsen *et al.* [24] proposed a deep invertible architecture, i-RevNet, which learns a one-to-one mapping between image and its feature. These networks achieve impressive accuracy on ILSVRC-2012 [29]. For experimentation, we choose bijective i-RevNet that takes images of size $224 \times 224 \times 3$ as input and the corresponding feature is of size $3072 \times 7 \times 7$. We use the pretrained i-RevNet provided in the official implementation[8] to test our proposed attack, OIMO. We randomly choose 100 correctly classified images belonging to 41 different classes from the validation set of ILSVRC-2012. Furthermore, we choose a starting image, I_{start}, belonging to a different class. We also restrict the search space for adversarial images using the clipping function $\text{Clip}_{I_{\text{start}},\epsilon}$ (i.e., the adversarial noise is clipped to ensure that the adversarial image I_{adv} will lie in an ϵ ℓ_∞-neighborhood of I_{start}). Starting with $I_{\text{start}} \in [0, 255]^{224\times 224\times 3}$, we run the proposed attack, OIMO, in order to mimic the feature for 100 images. Table V shows the success rate for different values of ϵ. The high success rate shows that the proposed attack can be applied for invertible architecture such as i-RevNet as well. This is because i-RevNet, despite being invertible, assigns similar features to dissimilar images. Fig. 7 shows one such successful adversarial example.

TABLE V

SUCCESS RATE OF OIMO FOR I-REVNET

ϵ	Success Rate
2	86.0 %
5	99.0 %
10	100.0 %

IV. QUANTITATIVE STUDY OF ADVERSARIAL NOISE

Table VI shows the peak signal-to-noise ratio (PSNR) for OIMO and task-specific methods. The PSNR is calculated as follows:

$$\text{PSNR} = 20 \log_{10}\left(\frac{255.0}{\sqrt{\text{MSE}}}\right)$$
$$\text{where} \quad \text{MSE} = \frac{\|I_{\text{adv}} - I_{\text{start}}\|_2^2}{m \times n \times 3} \tag{8}$$

where $I_{\text{adv}}, I_{\text{start}} \in [0, 255]^{m\times n\times 3}$.

[8] https://github.com/jhjacobsen/pytorch-i-revnet

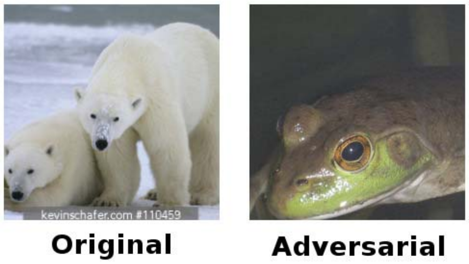


Fig. 7. Both the images are classified as ice bear by bijective i-RevNet.

TABLE VI
PSNR BETWEEN I_{adv} AND I_{start} FOR OIMO AND TASK-SPECIFIC METHODS

Model	Attack	PSNR (mean ± std)
Show and Tell	Show-and-Fool [5]	52.5 ± 6.7
	OIMO	23.8 ± 0.6
Show Attend and Tell	SSVM [28]	42.1 ± 1.2
	EM [28]	40.4 ± 0.9
	OIMO	26.1 ± 1.1
N2NMN	Xu et al. [6]	33.8 ± 3.7
	OIMO	27.6 ± 0.5

TABLE VII
SSIM BETWEEN I_{adv} AND I_{org} FOR MAF AND OIMO

Model	Attack	SSIM (mean ± std)
Show and Tell	MAF	$1.8 \times 10^{-4} \pm 1.3 \times 10^{-3}$
	OIMO	$6.1 \times 10^{-4} \pm 2.9 \times 10^{-3}$
Show Attend and Tell	MAF	$7.5 \times 10^{-4} \pm 2.7 \times 10^{-3}$
	OIMO	$6.8 \times 10^{-4} \pm 4.2 \times 10^{-3}$
N2NMN	MAF	$5.6 \times 10^{-4} \pm 1.5 \times 10^{-3}$
	OIMO	$4.5 \times 10^{-4} \pm 2.2 \times 10^{-3}$

From Table VI, it is evident that the PSNR is low for OIMO in comparison with other task-specific methods. This is mainly because task-specific methods can exploit the deficiencies of encoder as well as the decoder and such attack methods can be stopped at the exact instant when an adversarial image leads to the desired output. Agnosticity, in any form, generally leads to more noise. As an example, image-agnostic universal adversarial perturbations (UAPs) [30] are quasi-perceptible instead of being imperceptible. Table VII shows the SSIM [31] values between I_{adv} and I_{org} for the proposed methods. The near-zero values of SSIM clearly show that there is no resemblance between the original and adversarial images.

V. CONCLUSION AND FUTURE WORK

In this brief, we proposed a task-agnostic adversarial attack, MaF. The proposed attack exploits the noninvertibility of CNN-based feature extractors and is based on the hypothesis that if two images are indistinguishable for the feature extractor, then they will be indistinguishable for the model as well. The high success rate of MaF for three models across two tasks validates this hypothesis. We also show that the proposed attack works regardless of the depth of the feature extractor. Due to the task-agnostic nature, we need to run the attack only at the image level, which is a huge advantage in terms of time saved for tasks involving multiple modalities as input. We further propose a variant of MaF, named OIMO, which generates natural-looking adversarial images. The results for this variant of the attack show that it is possible to mimic features of an arbitrary image by making minimal changes to a fixed image. This is an important insight into the nature of CNN-based feature extractors. We also demonstrate the applicability of the proposed attack for invertible architectures such as i-RevNet.

As part of future work, from an attack perspective, one can explore different task-agnostic strategies which will work successfully with just the pretrained weights of the feature extractor. We found that using pretrained instead of fine-tuned weights leads to a drop in the success rate of the proposed attack. From a defense perspective, we show that invertible architectures, such as iRevNet, are not robust to the proposed attack. Hence, one can explore different feature extractors which are resistant to the proposed attack. If successful, one can use these feature extractors to develop end-to-end systems and check their robustness to task-agnostic as well as task-specific attacks.

REFERENCES

[1] S. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, “Shapeshifter: Robust physical adversarial attack on faster R-CNN object detector,” in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, vol. 11051, 2018, pp. 52–68.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–12.

[3] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in *Proc. ICLR Workshop*, 2017, pp. 1–13.

[4] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, “Adversarial examples for semantic segmentation and object detection,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1378–1387.

[5] H. Chen, H. Zhang, P.-Y. Chen, J. Yi, and C.-J. Hsieh, “Attacking visual language grounding with adversarial examples: A case study on neural image captioning,” in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2587–2597.

[6] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song, “Fooling vision and language models despite localization and attention mechanism,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4951–4961.

[7] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proc. ACM Asia Conf. Comput. Commun. Secur. (ASIA CCS)*, 2017, pp. 506–519.

[8] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017.

[9] Y. Shi, S. Wang, and Y. Han, “Curls & whey: Boosting black-box adversarial attacks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6512–6520.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[11] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)* 2015, pp. 1–12.

[12] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[13] A. Dosovitskiy and T. Brox, “Inverting visual representations with convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4829–4837.

[14] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5188–5196.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates, Inc., 2012, pp. 1097–1105.

[16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[17] K. Xu *et al.*, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 2048–2057.

[18] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, “Learning to reason: End-to-end module networks for visual question answering,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 804–813.

[19] Y. Huang, A. W. Kong, and K. Lam, “Attacking object detectors without changing the target object,” in *Proc. 16th Pacific Rim Int. Conf. Artif. Intell.* in Lecture Notes in Computer Science, vol. 11672. Cham, Switzerland: Springer, 2019, pp. 3–15.

[20] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[21] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the V in VQA matter: Elevating the role of image understanding in visual question answering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6325–6334.

[22] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 311–318.

[23] S. Banerjee and A. Lavie, “Meteor: An automatic metric for MT evaluation with improved correlation with human judgments,” in *Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summarization*, 2005, pp. 65–72.

[24] J.-H. Jacobsen, A. W. M. Smeulders, and E. Oyallon, “I-revnet: Deep invertible networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–11.

[25] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015 pp. 1–15.

[26] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://keras.io

[27] S. Sharma, L. El Asri, H. Schulz, and J. Zumer, “Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation,” 2017, *arXiv:1706.09799*. [Online]. Available: https://arxiv.org/abs/1706.09799

[28] Y. Xu *et al.*, “Exact adversarial attack to image captioning via structured output learning with latent variables,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4130–4139.

[29] O. Russakovsky *et al.*, “ImageNet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[30] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 86–94.

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.