IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 4, 2019, accepted June 14, 2019, date of publication June 19, 2019, date of current version July 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923674

Template Matching Based on Geometric Invariance in Deep Neural Network

YAMING CAO[1,2], ZHEN YANG[1], HAIJIAO WANG[3], XIAODONG PENG[1], CHEN GAO[1,2], AND YUN LI[1]

[1]National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Dharma Institute, Alibaba Cloud Intelligence Group, Ali Group, Beijing 100102, China

Corresponding author: Zhen Yang (yangzhen888@sina.com)

This work was supported in part by the Strategic Priority Program on Space Science of Chinese Academic Science under Grant XDA15014700.

ABSTRACT Machine learning models, especially deep neural networks (DNNs), have achieved state of the art in computer vision and speech recognition. However, with wide applications of DNNs, some problems have appeared, such as lack of interpretability and vulnerable to adversarial examples. Whether the judgment of the model is consistent with that of human is a key to the wide application and development of neural networks. In this paper, we propose a novel and interpretable method to enable the model to make the same judgment as humans in the adversarial examples, which is based on the geometric invariance between images of the same category. Template matching is combined with convolution neural network during the training and testing stage. Moreover, we manage to give a theorical proof. The geometric invariance features got from the template matching are fused with the features extracted by the convolutional layers. The experimental results demonstrate the temp_model (network added the template matching) has a higher test accuracy both on benchmark sequences and adversarial examples, and we use a visual method to explain the reason why adding template can make the network perform better. The generality and convergence of the network improve without increasing the model size and training time after adding the template as common sense.

INDEX TERMS Deep neural network, geometric invariant, interpretability, template match, adversarial example.

I. INTRODUCTION

Machine learning has excellent performance in computer applications. In particularly, since 2012, Hinton proposed Alex Net and got much better result than previous machine learning methods in classification on the ImageNet [1], and the neural networks have developed rapidly until now. Neural networks have achieved state of the art in many tasks, such as computer vision, natural language process, and speech recognition [2], [3].

However, some problems have also been exposed as the DNNs develop rapidly. For example, after adding some artificial perturbations to the input sample, the convolutional neural network makes a wrong classification with great confidence [4]–[8] at test time, and such perturbations also causes classification errors of other networks [5], [7]. Goodfellow proposed adversarial examples to describe the samples added with perturbations and explained the reason for the existence of the adversarial sample was weakness and linear nature of the neural network itself [7]. And adversarial examples have not only existed in computer vision, but also in malware detection [9], speech recognition [10], and semantic understanding [11]. Even in physical world for machine learning systems using signals from cameras or other sensors as input, they were vulnerable to adversarial examples [12]. Athalye *et al.* [13] used 3D printing to synthesize robust adversarial examples under different angles and illumination conditions and the results demonstrated the existence of 3D adversarial examples in the physical world. And Goodfellow found the same adversarial examples fools more than one model across architectures and training sets even if the adversarial had no access to the underlying model [7], [12]. Therefore, the problem of adversarial example is widespread in the field of matching learning.

There are many defense methods that used to solve the adversarial examples problem in the DNNs. A straightforward

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

and simple method is to detect the input sample and determine whether the input sample is adversarial example by the statistical distribution rule of the sample [14]. Some researchers want to eliminate the interference of adversarial examples by purifying and filtering the input [15]. However, these simple detection and purification methods are difficult to work with in many complex scenarios and models. It is an effective defense method that enhancing the robustness of the model itself through adversarial training [16]. But it is actually difficult to design a perfect adversarial training before the attack. And with the wide application of deep neural networks in the fields of finance, medical care and autonomous driving, the problem that deep neural networks cannot clearly explain their decision-making behavior has attracted more and more attention [5], [17]. The discovery of adversarial examples makes the interpretability of the network more meaningful. The main reason why neural network is not interpretable is that it lacks common sense and is vulnerable to adversarial example interference. Adding common sense to the neural network is an effective means to make the application of the neural network wide and more stable. However, there are currently many difficulties in integrating common sense and knowledge into the field of neural networks [18], [19]. Many common senses in human cognition are highly abstract. How to quantify common sense and integrate with the high-dimensional features learned by neural network needs to be solved.

In this paper, we propose a novel method of combining template matching [20], [22] with convolutional neural networks to integrate common sense into neural network. We add the geometric invariant features of the image extracted by the template matching to the neural network as common sense to make the new model can better resist the disturbance and be more interpretable. The production of the adversarial examples is basically to add some artificially designed noise to the original image. The human can see more or less the change of the image, but can still make a correct judgment based on the valid information. However, this causes the convolutional neural network or other machine learning models to output error results. Convolutional neural networks or other machine learning modules focus on changes in each pixel and be deceived by the adversarial examples. These noises do not affect human extracting valid features of the images. The geometric invariant is the part of reason why human can judge correctly without the effect of designed noise.

Moments and invariants of images are very useful for invariant feature extraction due to their rotation, scale, and translation invariance properties [21]. Geometric invariance is one of the basic common senses and is used in many object recognition and image matching [22], [23]. In many 2d and 3d cases, the geometric invariant can be used for the preservation of cultural heritage [22], feature extraction of medical images [23], and reconstruction of images [24].

We deem that the geometric invariant may be used to reduce or eliminate the influence of adversarial noise, and improve the network's interpretability for classification results. And the thinking is validated using the MNIST dataset [25] and LeNet-5 [26] under the Caffe [27] framework. The normal input and adversarial examples are respectively calculated with the templates to get many normalized correlation coefficients, and the vector is obtained from the max normalized correlation coefficient calculated from each template. This vector is added to the last fully connected layer of the convolutional neural network to obtain the extended vector, which is then fed to the SoftMax layer to output the recognition result. In the process of back propagation, no gradient calculation is performed for the template matching process. In general, many features based on geometric invariance can be extracted by template matching to complement the recognition of convolutional neural networks. In the process of training, the network with template matching and the network without it are trained on the normal data set and tested on normal and adversarial data sets. We demonstrate the network with the template matching is indeed more effective by comparing the experiment results, and the training time and model size. And experimental results reveal a substantial rise of classification scores in adversarial set. In benchmark sequences, the network with the template matching performed better. The results of experiments are consistent with the theoretical derivation. In the end, we use a visual method [28] to explore the difference of internal neurons between the temp_model and without_temp_model (LeNet-5), and find that the reason why adding template can make the network perform better is reasonable. We also explore the performance of temp_model and without_temp_model in adversarial training. The contributions of this study are as follows:

1) First, we propose a novel and reasonable method to add the template matching in DNNs as a common sense and use the geometric invariance to improve the interpretability and generality of the network.
2) Second, we verify the effectiveness of the method under normal dataset and adversarial examples, including adversarial training, and use the visual method to explore the enhancement of signal feature map after adding the template matching.
3) Third, our method also proposes a new way in some model applications with a special focus on security, which is that adding an offline template or other common sense like that can increase the security of the model and resist malicious attacks.

The remainder of this paper is organized as follows. First, we introduce the method in Section II, including dataset and methodology. We then present the theorical proof to establish the experiment in Section III. We show the experiment results and analysis it using a visual method in section IV. Finally, we conclude the paper in section V.

II. METHOD

A. INTRODUCTION OF MNIST

The MNIST database of handwritten digits have a training set of 60,000 examples, and a test set of 10,000 examples [25].

The digits have centered in a fixed-size image. And it is a good database for people who want to learn machine learning and pattern recognition techniques without spending much time preprocessing and formatting.

B. METHODOLOGY

Template matching is a basic and original method in digital image processing [20]. The principle of this method is very simple, traversing every possible position in the target image, comparing whether the place is similar to the template, including the geometric invariance, and when the similarity is high enough, we think that our target is found. In the process of comparing the template with the original image, there are many algorithms to measure the similarity of the match. Standard correlation matching (CV_TM_CCOEFF_NORMED) is the most complex similarity algorithm supported by OpenCV. Geometric similarity is the main manifestation of similarity between images with fixed structural features in grayscale.

The shape of template is $w \times h$, which w and h mean the width and height of the template image. And the shape of input image is $m \times n$. m and n stand for the width and the height of input image. $T(i,j)$ and $I(i,j)$ stand the value of pixel (i,j) of template and input image. The template slides as a window across the entire image in steps of one pixel. i' and j' are the cumulative number of steps the window has crossed. Each time it is slid, it will be calculated according to (4) to obtain the corresponding correlation coefficient $R(i,j)$ to measure the geometric invariance between template and the same size part of image. At the end of the sliding, there is a $(m - w + 1) \times (n - h + 1)$ matrix consisted of all the correlation coefficients. The max element of the matrix is chosen to measure the geometric invariance between the input image and template.

The calculated correlation coefficient is positive with the geometric invariance between the target image and the template. The same template can also be used to obtain the maximum of the corresponding correlation coefficients after the image is added to the perturbation to be the adversarial example. We use formula (6) to calculate the r to measure the geometric invariance between the correlation coefficient of normal image and that of the adversarial example. Under the premise of geometric invariance, r is very close to 1, which is the theoretical basis for the temp_model performing better under the adversarial examples. This theoretical basis is proved in the section III.

III. ESTABLISHMENT OF EXPERIMENTS

Before and after image plus adversarial noise, the correlation coefficient of it and template does not much change due to the geometric invariant. These invariant features are effective to improve the recognition accuracy of model to the adversarial examples. The templates used to extract the invariant features of adversarial examples are shown as Fig. 1.

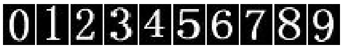


FIGURE 1. **The templates used to match. Each template is 20 × 25 pixels.**

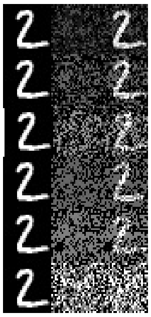


FIGURE 2. **Adversarial examples produced by FGSM with different $\varepsilon(0.1 - 0.6)$, and ε increased 0.1 by line. In each line, the left side is the normal examples, the middle is the designed noise, and the right side is the adversarial examples.**

There are many ways to produce perturbation to get adversarial samples. In this paper, the adversarial samples used in the experiment were produced with the Fast Gradient Sign Method (FGSM) [12] algorithm by using (1). ε is the perturbation added on the normal input x and it is small enough so that human can still make correct judgment on the adversarial example $\widetilde{x}$. But with the whole image $(m \times n)$ and the weight vector of w, the activation grows more than εmn. As a result, the normal image with perturbation is misclassified by neural network. The perturbation is not random noise, and it is calculated by some special algorithm. The loss function used to train model is $J(\theta, x, y)$. *sign*() is the sign of the derivative of $J(\theta, x, y)$, 1 or -1. Fig.2 shows adversarial examples produced by FGSM under different ε.

$$\widetilde{x} = x + \varepsilon * sign(\nabla_x J(\theta, x, y)) \tag{1}$$

The method of template matching can measure the geometric invariance between image and template. Specifically, in addition to subtracting their means, there are also divided by their respective variances. The normalization operation on template $T(i,j)$ and image $I(i,j)$ can be seen as (2) and (3). Our image to be inputted and the template are standardized after the two steps, so that the illumination of the target image and the template can be changed alone without affecting the calculation result.

And from the expression of the correlation coefficient $R(i,j)$ as (4), the similarity of template $T(i,j)$ and image $I(i,j)$, the perturbation added in the every pixel of the image can be weaken or eliminated in some extent with the operation subtracting means and dividing respective variances.

$$T'(i,j) = T(i,j) - \frac{1}{w \times h} \sum\nolimits_{i',j'} T(i',j') \tag{2}$$

$$I'(i,j) = I(i,j) - \frac{1}{m \times n} \sum\nolimits_{i',j'} I(i',j') \tag{3}$$

$$R(i,j) = \frac{\sum_{i',j'} (T'(i,j) * I'(i+i',j+j'))}{\sqrt{\sum_{i',j'} T'(i,j)^2 \sum_{i',j'} I'(i+i',j+j')^2}} \tag{4}$$

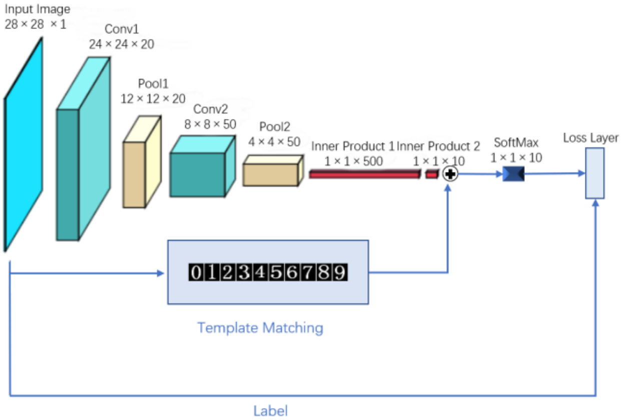


FIGURE 3. The structure of the changed Lenet-5 added template matching, the New-LeNet.

TABLE 1. Accuracy of two models under different training iteration times: (a) under normal MNIST data sets; (b) under Caffe-Ocr data sets. Temp_ac means the accuracy of the New-LeNet, and without_temp means the accuracy of LeNet.

(a)

iterations	2000	4000	6000	8000	10000
temp_ac	0.9882	0.9899	0.9910	0.9922	0.9933
without_temp	0.9866	0.9891	0.9911	0.9907	0.9915

(b)

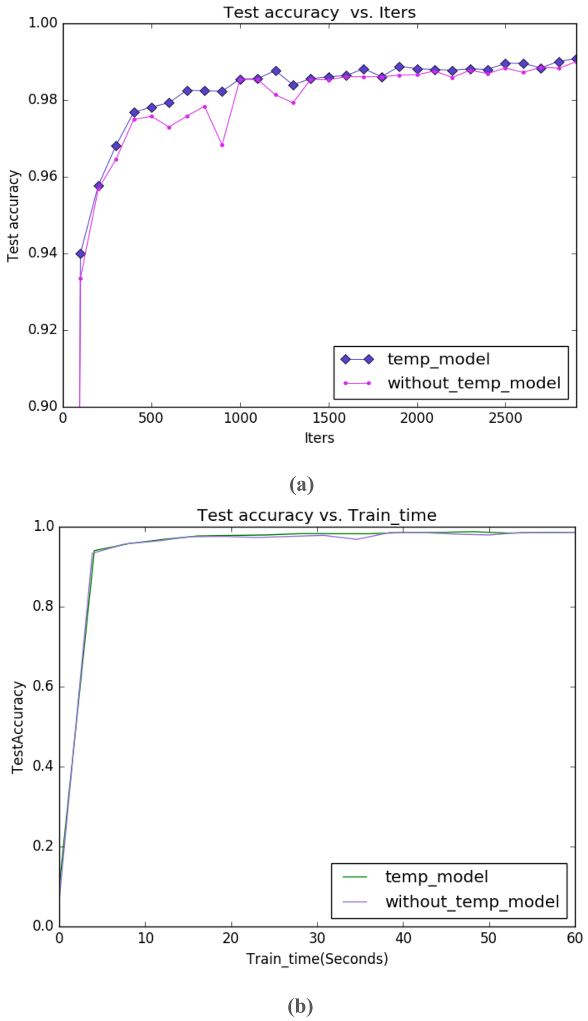
iterations	0	100	200	300	400
temp_ac	0.0171	0.0496	0.2492	0.8408	0.999
without_temp	0	0.0005	0.1237	0.7621	0.9948

Each pixel in the image can be thought of as a number in a sequence. The input image $I(i,j)$ is expressed as the sequence $A(a_{11}, a_{12} \cdots\cdots a_{mn})$, and the template is $T(t_{11}, t_{12} \cdots\cdots t_{wh})$ and the perturbation is $B(b_{11}, b_{12} \cdots\cdots b_{mn})$. a_{ij}, t_{ij} and b_{ij} stand for the value of pixel (i,j) of input image, template and the perturbation. And every element of A and B has the following conditions:

$$a_{ij} \subseteq [0, 255], \quad b_{ij} \subseteq \{-\varepsilon, +\varepsilon\} \tag{5}$$

The means and variances of A are E_A and D_A, and those of B are E_B and D_B. The max correlation coefficient between template and adversarial sample is R_{adv-t}, and that between template and normal image is R_{i-t}. R_{i-t} is calculated using the normal image and the template, which corresponds to the label of the normal image. The way to get R_{adv-t} is the same as R_{i-t}. We use r to measure the relationship between R_{i-t} and R_{adv-t} to reflect the geometric correlation between the adversarial example and normal image. From formula (4), we can get R_{i-t} and R_{adv-t}, and then we can get r:

$$r = \frac{R_{i-t}}{R_{adv-t}} = \frac{\sqrt{D_A + D_B} * \sum_{j}^{n} \sum_{i}^{m} (a_{ij} - E_A)\, t_{ij}}{\sqrt{D_A} * \sum_{j}^{n} \sum_{i}^{m} (a_{ij} + b_{ij} - E_A - E_B)\, t_{ij}} \tag{6}$$



(a)

(b)

FIGURE 4. Results of test in normal dataset. (a) Test accuracy of two models during training. Temp_model means the New-LeNet, and without_temp_model means the LeNet, without template matching. (b) The test accuracies of two models changes with train time.

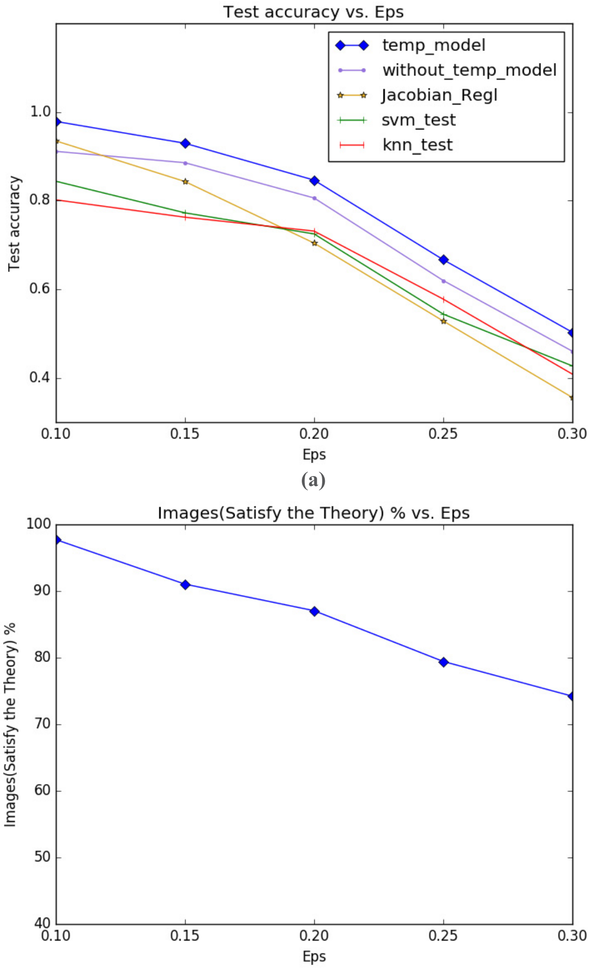
TABLE 2. Accuracy of all models under different adversarial examples of perturbations (eps).

eps	0.1	0.15	0.2	0.25	0.3
temp	0.9787	0.9294	0.8462	0.6666	0.5028
without_temp	0.9111	0.8855	0.8059	0.6196	0.46
Jacobian_Regl	0.9348	0.8431	0.7041	0.5283	0.3556
svm_test	0.8437	0.7724	0.7248	0.5438	0.4269
knn_test	0.8017	0.7625	0.7312	0.5777	0.4084

According to the simplification of the 6 and the limitation of (5), we can get:

$$1 \leq r < \sqrt{1 + \frac{D_B}{D_A}} \quad \text{and } 0 \leq D_B \leq \varepsilon^2 \tag{7}$$

Most of time the ε is not more than one because of the restriction of L_∞ norm [12]. So, the D_B is far less than D_A, and in theory the correlation coefficient between the image



(a)

(b)

FIGURE 5. **Results of test in adversarial examples. (a) Test accuracy of for models during testing under adversarial examples of different perturbations. (b) The percentage of images under different eps, which the correspond *r* of it is closest to 1 and the label of it is the same with the template label.**

and the template does not change substantially before and after the appropriate disturbance added on the image. We can see that the adversarial examples retain some of the geometric invariant features of the original images. Based on intuition and above theoretical support, we design an improved network as Fig. 3 that combines template matching with the neural network.

The temp_model consists of two parts: LeNet and template matching. The input image is paralleled through neural network (LeNet) and template matching. In the out of the second inner product layer of the LeNet, a feature vector of $1 \times 1 \times 10$ shape is obtained to represent the high-dimensional features extracted in the previous layers. At the same time, the image is matched with ten templates using the standard correlation matching algorithm, and ten matrixes of $(m - w + 1) \times (n - h + 1)$ shape are obtained. The maximum value in each matrix is used to represent the similarity between the image and the template, so a vector of $1 \times 1 \times 10$ shape is obtained. This vector and the vector of the result of the second inner product layer are integrated into one feature vector by weighting as input of SoftMax layer to output the recognition result. And the geometric invariance features extracted by template matching are also added into the neural network after the two vectors are integrated into one vector. No gradient calculation is performed for the template matching process in order to keep the template as a common sense invariant, and the training and testing of the temp_model are the same as the ordinary CNN models. In the process of training, the geometric invariance features extracted by template matching are used to make the neural network converge faster by the gradient back of the recognition result.

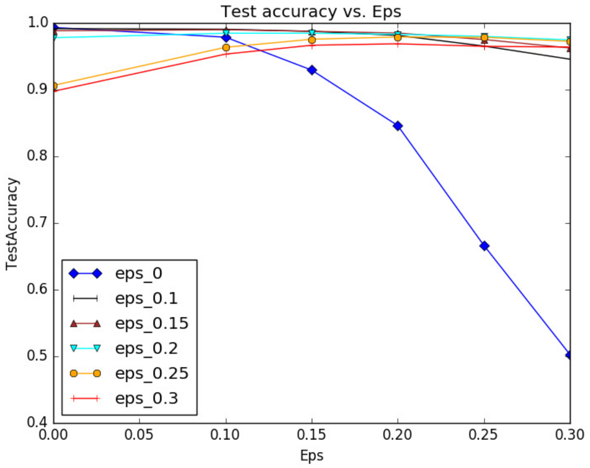


FIGURE 6. **The test accuracy of temp_model under different eps of adversarial test dataset. And Each line represents the data set under which the model is trained.**

IV. EXPERIMENT

To verify the validity of the neural network with template matching from an experimental perspective, we conduct three experiments. In each experiment, each model is guaranteed to be consistent over the hyperparameters. The model structure is shown in Fig. 3, which is added template matching with the LeNet.

A. TEST IN NORMAL DATASET

First experiment is the New-LeNet and LeNet are trained and tested in the normal dataset with the same hyperparameters. We also used a data set of 26 uppercase English letters, caffe-ocr, to test the temp_model and without_temp_model. Each letter of the dataset was randomly generated from 10 Word formats such as AndaleMono, Arial, ArialBlack, and TimesNewRoman. The two models are trained separately in the case of different training rounds, and then the two models are tested on the normal test set. And we can find that in the case of almost all training times from the Table 1 and Fig. 4 (a), the accuracy of New-LeNet in MNIST and Caffe_Ocr is higher than that of the LeNet. The former is also more stable and less oscillating during the training process. The template may have a certain guiding effect in the model, which makes the model converge faster when training.

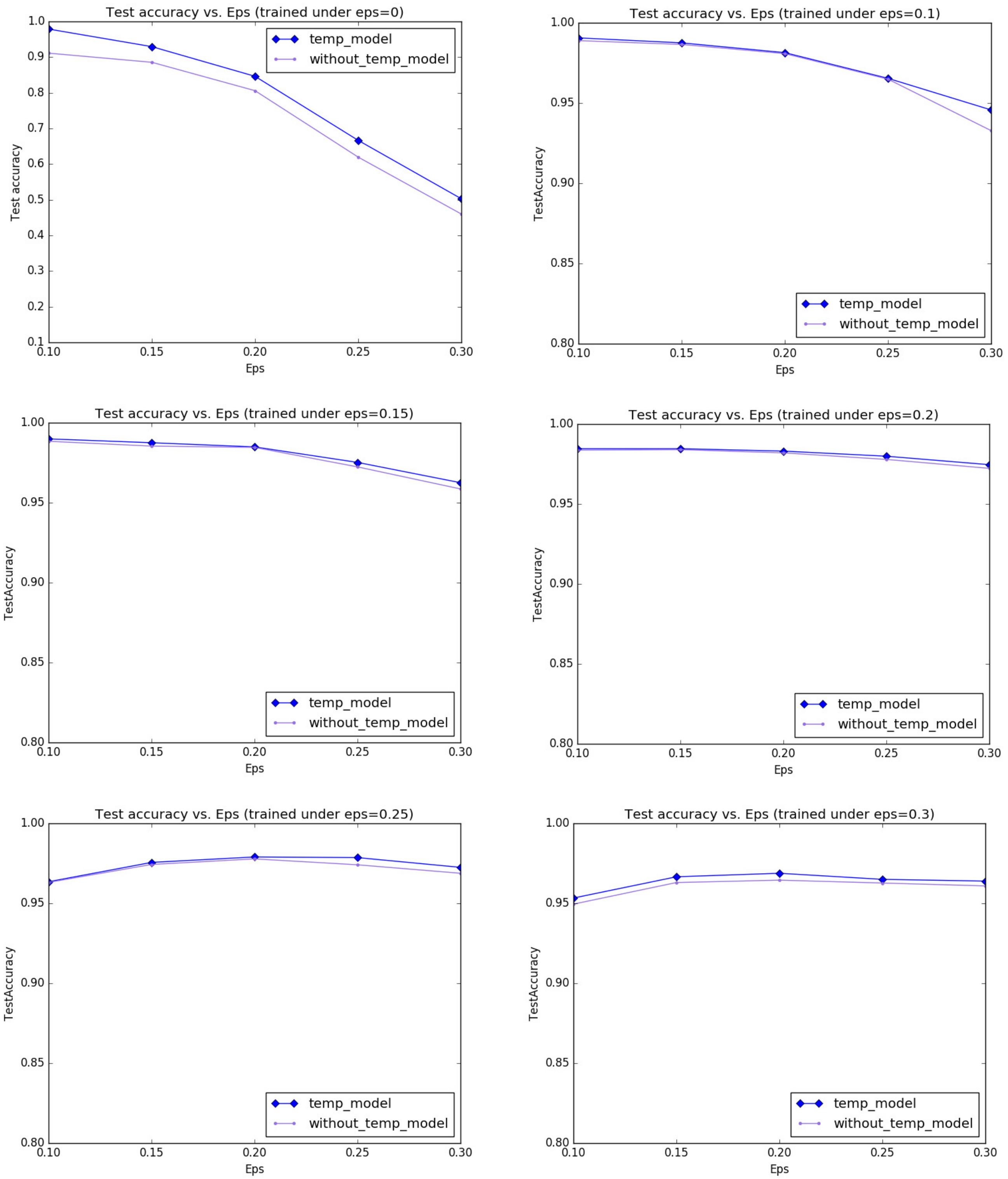


FIGURE 7. The test accuracy of temp_model and without-temp_model tested under different eps of adversarial examples. And the different graphs represent that the two models are trained under different eps of adversarial dataset.

The accuracy of the two models is close at some points, and the accuracy of temp_model is even slightly lower than that of the without_temp_model when the number of iterations is 6000 in Table 1. The reason is the local micro-oscillation caused by the step limitation of the neural network during training process. And from the quantitative

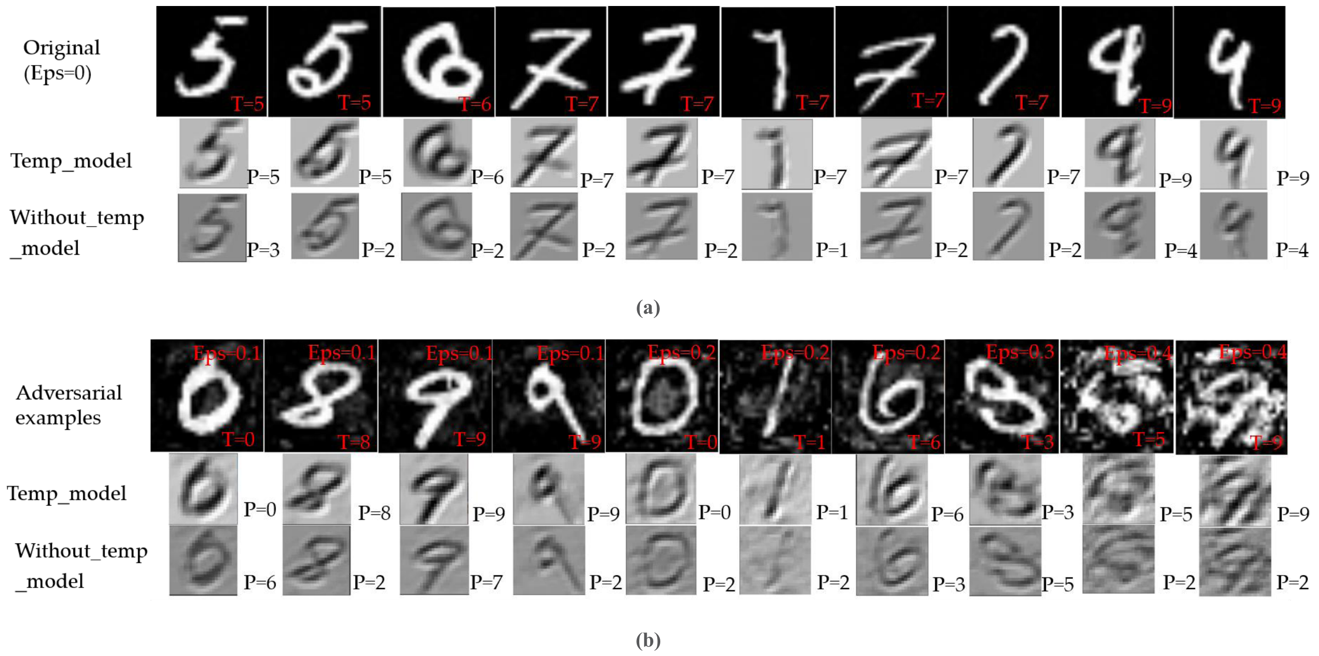


FIGURE 8. Visualization of the same channel of the same convolutional layer. (a) Visualization of feature map of images from the test dataset without any noisy, the value of T is the label of the image and the value of P is the prediction of two models. (b) Visualization of feature map of adversarial examples under different eps, the value of Eps represents how much noise the image is generated under.

analysis of Table 1, the accuracy of the temp_model is on average 0.14% higher than that of the without_temp_model. So, it is accepted as a local micro shock that the accuracy of temp_model is 0.01% lower than that of the without_temp_model. Fig. 4 (b) shows that the test accuracies of the two models are basically the same as the change of training time and adding template does not increase the extra training time. And both models are the same size (1.7M).

B. TEST IN ADVERSARIAL EXAMPLES

The second experiment is to test the temp_model, without_temp_model, SVM and KNN under the adversarial examples, which were trained under the normal dataset. And the adversarial examples are produced with FGSM under different perturbations ε (Eps). The results of the Jacobian Regularization were added as a comparison under the same Eps, which was used to improve DNN robustness to adversarial attacks [29]. As the noise continues to increase, the picture becomes more and more blurred, and the accuracy of all models declines. However, as Table 2 and Fig. 5 (a) show, no matter how much disturbance, the accuracy of the model with template matching, temp_model, is higher than other models.

Fig. 5 (b) shows that the images that satisfies the theory under different eps account for the percentage of the classifying correctly images. Satisfying the theory means that the template label is the same with the image label and the correspond r is closest to one, compared to other r calculated using other templates. Experimental results suggest that the invariant features extracted by template matching do improve the generality of the network in the adversarial examples.

C. ADVERSARIAL TRAINING AND VISUALIZION

As [16] says, adversarial examples can reduce the test error if they are used for adversarial training models. So, the final experiment is to train the two models under the adversarial examples dataset and then look at their performance on the normal test dataset and adversarial examples test dataset. And we try to use a visual method [28] to explore the difference of internal neurons between the two models, and find that the reason why adding template can make the network perform better is reasonable.

It can be easily seen that adversarial training can greatly improve the robustness of the model to the adversarial examples from Fig. 6, and the models training under different eps of adversarial examples are also different in robustness. There may be a specific noise value making the model performing best in the adversarial training of all eps. And the performance of temp_model is better than that of without_temp_model from Fig. 7, under every eps of adversarial examples trained.

Fig. 8 shows the visualization of feature map of normal dataset and adversarial examples, and the same channel of the first convolutional layer is chosen. As seen in the figure, because the templates are added to the temp_model, the feature map of different models looks slightly different. However, with all images from normal dataset and adversarial examples, the feature maps from temp_model are brighter and clearer than those from without_temp_model. Fig. 8 (a) shows the effective features in the feature map of temp_model get more enhancements from the background, and the geometric features in the image are better preserved than that of without_temp_model.

Fig. 8 (b) shows the difference of visualization of feature map for temp_model and without_temp_model. Images are generated under different eps. As seen in the figure, temp_model can extract the effective geometric features of image more comprehensively under the same level of lightly noise, without being covered by noise. However, temp_model can no longer extract the geometric features from the noise when the adversarial noise continues increase (Eps = 0.4 or bigger), just making both of them stand out in the background. When the adversarial noise is as large as the last two images, even human cannot accurately identify the numbers in the images. And the results confirm the theoretical derivation of the model (Secretion 3) and Fig. 5.

V. CONCLUSION

In this paper it can be easily seen that the adversarial examples have a big impact on the accuracy of the model, especially if the model is not considering the adversarial training during the process of training. And we proposed a novel method to improve the consistency of the judgment between the network and humans on the slightly adversarial examples, which combined convolution neural network with template matching based on the geometric invariant. The geometric invariant from the template matching is indeed effective under the adversarial examples and also improves the interpretability of the model, and the model is more stable during the training process. There is basically no additional time and space overhead after adding template matching. And the proposed method is not limited by computing resources when applied. Adversarial training can greatly improve the generality of the model, and the temp_model have a better performance than without_temp_model in adversarial training of all eps. There may be a specific noise value making the model performing best in the adversarial training of all eps. Finally, the same feature map of the two models at the same input is compared by visual method, and the improvement of the model after adding the template is qualitatively seen.

This method also proposes a new way of thinking. In some model applications with a special focus on security, consider adding an offline template to increase the security of the model, and template is similar to a graphical map of knowledge. On the one hand, this method can improve the accuracy of the model, on the other hand, it can also improve the resistance of the model to the adversarial examples. In future work, we consider adding more common sense and reasoning to the neural network to make the model more stable and humans more likely to understand the decision making of the model.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst (NIPS)*, 2012, pp. 1097–1105.

[2] Y. Taigman, Y. Ming, M. A. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1701–1708.

[3] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[4] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 427–436.

[5] J. Bruna, C. Szegedy, I. Sutskever, I. Goodfellow, W. Zaremba, R. Fergus, and D. Erhan, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–10.

[6] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli "Evasion attacks against machine learning at test time," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 387–402.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–11.

[8] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–24.

[9] W. Hu and T. Ying, "Generating adversarial malware examples for black-box attacks based on GAN," 2017, *arXiv:1702.05983*. [Online]. Available: https://arxiv.org/abs/1702.05983

[10] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 1–7.

[11] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," 2017, *arXiv:1707.07328*. [Online]. Available: https://arxiv.org/abs/1707.07328

[12] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, *arXiv:1607.02533*. [Online]. Available: https://arxiv.org/abs/1607.02533

[13] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," 2017, *arXiv:1707.07397*. [Online]. Available: https://arxiv.org/abs/1707.07397

[14] L. Xin and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 5764–5772.

[15] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," 2017, *arXiv:1711.00117*. [Online]. Available: https://arxiv.org/abs/1711.00117

[16] F. Tramèr, A. Kurakin, N. Papernot, B. G. Dan, and P. Mcdaniel, "Ensemble adversarial training: Attacks and defenses," 2017, *arXiv:1705.07204*. [Online]. Available: https://arxiv.org/abs/1705.07204

[17] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. Mcdaniel, "The space of transferable adversarial examples," 2017, *arXiv:1704.03453*. [Online]. Available: https://arxiv.org/abs/1704.03453

[18] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in *Proc. World Wide Web Conf. World Wide*, 2018, pp. 1835–1844.

[19] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The 'something something' video database for learning and evaluating visual common sense," 2017, *arXiv:1706.04261*. [Online]. Available: https://arxiv.org/abs/1706.04261

[20] A. L. Ratan and W. E. L. Grimson, "Training templates for scene classification using a few examples," in *Proc. IEEE Workshop Content-Based Access Image Video Libraries*, Jun. 2012, pp. 90–97.

[21] S. P. Singh and S. Urooj, "Combined rotation-and scale-invariant texture analysis using radon-based polar complex exponential transform," *Arabian J. Sci. Eng.*, vol. 40, no. 8, pp. 2309–2322, 2015.

[22] S. Maweheb, S. Malek, and G. Faouzi, "Geometric invariance in digital imaging for the preservation of cultural heritage in Tunisia," *Digit. Appl. Archaeol. Cultural Heritage*, vol. 3, no. 4, pp. 99–107, 2016.

[23] S. Urooj and S. P. Singh, "Geometric invariant feature extraction of medical images using Hu's invariants," in *Proc. Int. Conf. Comput. Sustain. Global Develop.*, 2016, pp. 1560–1562.

[24] X. Wang, X. Chen, and S. Qu, "Three-dimensional geometric invariant construction from images," in *Proc. Int. Conf. Inf. Technol. Softw. Eng.*, 2013, pp. 335–343.

[25] Y. C. LeCun, C. J. C. Burges, and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," 2014, *arXiv:1408.5093*. [Online]. Available: https://arxiv.org/abs/1408.5093

[28] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Proc. Int. Conf. Mach. Learn.-Deep Learn. Workshop*, 2015, pp. 1–12.

[29] D. Jakubovitz and R. Giryes, "Improving DNN robustness to adversarial attacks using jacobian regularization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 525–541.

YAMING CAO received the B.S. degree in automation from the Beijing University of Chemical Technology, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences. His research interests include deep learning and interpretability of neural networks.

ZHEN YANG received the B.S. and M.S. degrees in communication and electronic engineering from National Defense University, Changsha, China, in 1994 and 1997, respectively, and the Ph.D. degree in computer application from the University of Chinese Academy of Sciences, Beijing, China, in 2014.

He is currently a Professor with the National Space Science Center, Chinese Academy of Sciences. His research interests include complex system simulation, spatial task collaborative design and demonstration, spatial information services, and distributed space systems.

HAIJIAO WANG received the B.S. and M.S. degrees in geographic information system from China Agricultural University, Beijing, China, in 2011 and 2013, respectively, and the Ph.D. degree in computer application from the University of Chinese Academy of Sciences, Beijing, China, in 2018.

He is currently a Research Assistant with the Dharma Institute, Alibaba Cloud Intelligence Group, Ali Group. His research interests include natural language processing and text processing.

XIAODONG PENG received the B.S. degree from the Xidian University of Electronic Technology, Xian, China, in 2002, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently a Professor with the National Space Science Center, Chinese Academy of Sciences. His research interests include space mission demonstration and simulation, satellite situation analysis and demonstration, massive data management and visualization, and scene perception and reconstruction.

CHEN GAO received the B.S. degree in mechanical engineering and automation from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2010. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences. His research interests include system engineering and simulation.

YUN LI received the B.S. and M.S. degrees in software engineering from Beihang University, Beijing, China, in 2014 and 2017, respectively.

He is currently a Research Assistant with the National Space Science Center, Chinese Academy of Sciences. His research interests include computer vision and robotics.