Supplementary Material on "Deep Learning from a Statistical Perspective"

1 Implementation: Image Classification

In this section, we implement CNN in image classification to explain the rationales behind the CNN, and also demonstrate illustrate the robustness issue arising from the CNN

In the first implementation, we use the pre-trained VGG-16 [4] for demonstration. TheVGG-16 is a deep CNN trained on ImageNet and provides a 92.7% top-5 class test accuracy in classifying 1000 image categories. The network consists of 13 convolutional layers, 3 fully-connected layers and 5 max-pooling layers with 138 million parameters. This deep network receives colored images of size $224 \times 224 \times 3$ as input, and provides the classification probability of each category as output for each image. Figure 1 shows the architecture of the VGG-16 and the size of each layer.

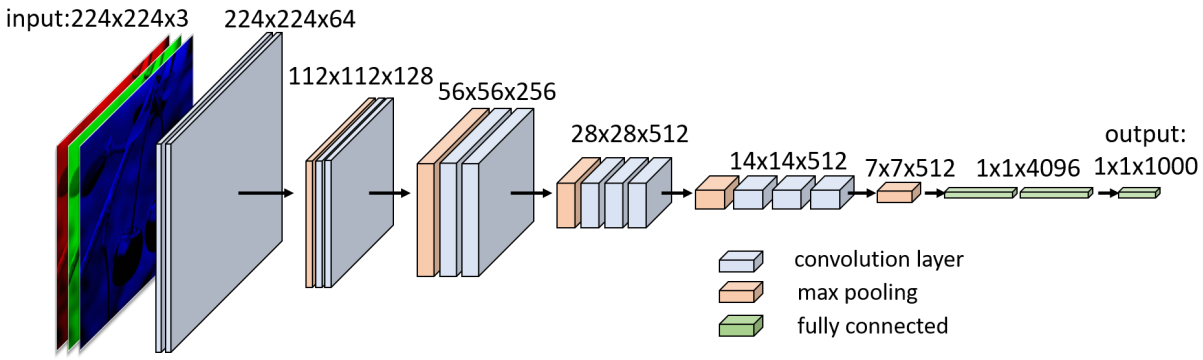

Figure 1: Architecture of the VGG-16. The numbers denote the size of each layer.

In the first two examples, we input the images of a sorrel horse and a panda, respectively. The VGG-16 correctly classifies the two images with a probability almost equal to 1. In addition, we use the Grad-CAM [2] to highlight the most salient pixels associated with the classification to provide a visual explanation. The heatmaps in Figure 2 generated by the Grad-CAM indicate that VGG-16 captures the entire shape of the horse and a region of panda's face successfully. For another example, we input an image which contains a tiger and a goat in the same image. As a result, the neural network classifies the image as a "tiger" by correctly identifying the stripes from the image.

Although the CNN misclassifies the goat as a lion with a small probability ($p = 0.03$), it correctly identifies the position of the goat.

These two examples demonstrate the interpretability of the features extracted by CNN and illustrates the similar mechanism between the CNN and the human vision.

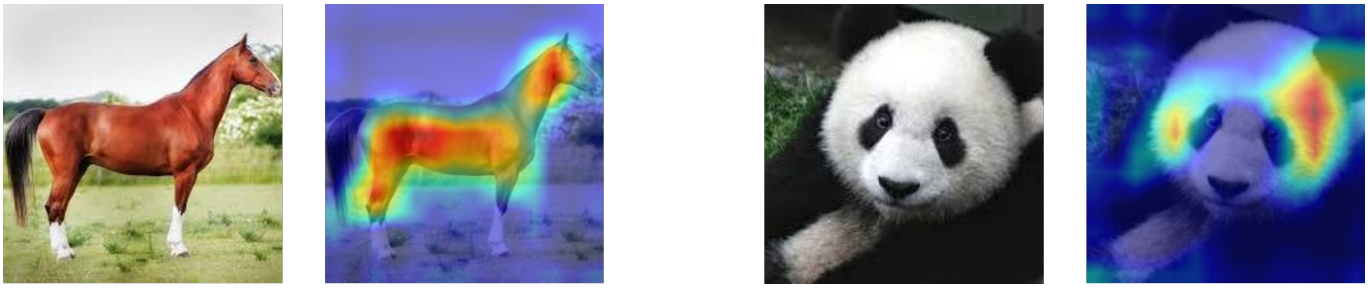

Figure 2: Classification results of a horse and a panda, each pair of images includes input image (*left*) and the corresponding heatmap (*right*) indicating the location of important pixels for classification. The warmer color in the heatmaps indicates higher pixel importance.

Figure 3: An image of a tiger and a goat is classified as a "tiger"($p = 0.71$) and a "lion"($p = 0.047$), showing the input image (*left*), the heatmap for "tiger" (*middle*) and the heatmap for "lion" (*right*)

For the second implementation, we demonstrate the robustness issue of the CNN through two handwritten digit datasets. The translated MNIST dataset [1] randomly translates the 28×28 handwritten digits on a black background of 40×40 pixels. And the affNIST dataset [5] adjusts the digit images through more complex affine transformation other than translation through rotation, scaling and shearing. Figure 4 provides examples from the translated MNIST dataset and affNIST dataset. We first train a two-layer CNN with 50,000 samples from the translated MNIST dataset and then test on 10,000 samples from the translated MNIST dataset and affNIST dataset, respectively.

The trained model achieves an accuracy of 98.81% on the training set and 98.26% on the translated MNIST testing set. However, it only results 65.97% accuracy on the affNIST testing set, which implies that CNNs are not robust against affine transformation. This phenomenon suggests that the CNNs may fail when the input data is highly heterogeneous. To address this issue, one can apply data augmentation [3] to increase the training set by adding more images with small distortions to the original images. However, data augmentation increases significantly on training cost and could still fail when the types of image transformation are not included in the training set.

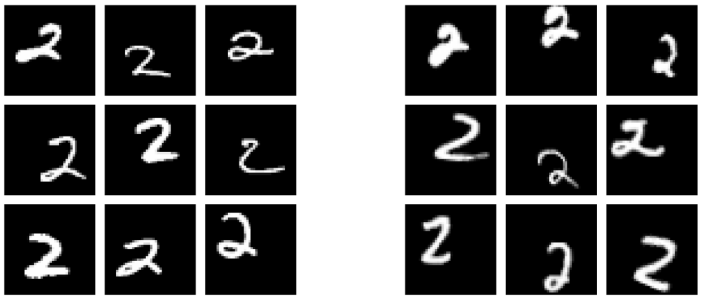

Figure 4: Samples of digit 2 from the translated MNIST dataset (*left*) and affNIST dataset (*right*). The translated MNIST applies only translation on MNIST images, and affNIST applies more complex affine transformation, including rotation, scaling and shearing.

References

[1] LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*.

[2] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626.

[3] Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.

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

[5] Tieleman, T. The affNIST dataset. http://www.cs.toronto.edu/ tijmen/affNIST.