ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Visualizing Image Classification in Fourier Domain

Florian Franzen and Chunrong Yuan

Autonomous Systems Lab
TH Köln — University of Applied Sciences
Betzdorfer Str. 2, 50679 Cologne, Germany
{florian.franzen, chunrong.yuan@th-koeln.de}

Abstract. Image classification is successfully done with Convolutional Neural Networks (CNN). Alternatively it can be done in Fourier domain avoiding the convolution process. In this work, we develop several neural networks (NN) for classifying images in Fourier domain. In order to understand and explain the behaviour of the built NNs, we visualize neuron activities and analyze the underlying patterns relevant for the learning and classification process. We have carried out comparative study based on several datasets. By using images of objects with partial occlusion, we are able to find out the parts that are important for the classification of certain objects.

1 Introduction

Since the introduction of Convolutional Neural Networks(CNN) by LeCun et al. [1], most systems for object recognition use CNN. The advantage of CNN is its invariance to translations and partial invariance under other kinds of transforms. This is mainly achieved by the convolution process. A convolution in the spatial domain can be substituted by a multiplication in the frequency domain. In this sense, Fast Fourier Transform (FFT) is used by Mathieu and LeCun for the fast training of CNN [2]. With the help of visualization, it was possible to explain, how CNN works [3][4].

FFT has also been used in the work of Chen et. al. [5], in the context of frequency sensitive hash nets (Freshnets). Their purpose was mainly the compression of CNN. In 2017, Fourier Convolutional Neural Networks (FCNN) was proposed, where classification was carried out entirely in Fourier domain [6]. While their work emphasizes the speed effect, our major goal is to gain more insight in understanding image classification in Fourier domain. By performing a backward process similar to a deconvolution in CNN [3], we visualize patterns which are learned by the networks. While trying to explain the classifiers which are built by the NN in Fourier Domain, we focus on information visualization as a role of exploration and confirmation, as is formulated by Frénay and Dumas [4].

2 Fourier Domain Method

In our work we use the Discrete Cosinus Transform (DCT) as a variant of the Fourier Transform. An image can be transformed into its DCT image using the following equation:

$$X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right] . \quad (1)$$

where k = 0,...,N-1 and N the number of pixels in horizontal direction (denoted by subscript 1) and vertical direction (denoted by subscript 2). An absolute value DCT image can be calculated by converting all values in a DCT image into positive ones, using

$$Y_{k_1,k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \left|\cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]\right| . \quad (2)$$

Shown in Fig. 1 from left to right are an input image, its DCT and absolute value DCT respectively. Note that the image intensity values were normalized for printing and visualization purpose, with DCT minimum shown as black and maximum as white.

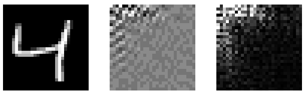

Fig. 1: Image of handwritten digit 4 and its DCT and absolute value DCT image.

In the DCT image, low frequencies are located in the upper left corner, while high frequencies are located in lower right corner. Usually the plain DCTs of the images have both negative and positive values, where a strong absolute value show activity at one frequency in the frequency domain. So for the purpose of classification, only the horizontal and vertical position in the DCT and the absolute values of the coefficients are important. During the training phase of NN, the weights of a neuron will be strengthened by the positive and weakened by the negative input. As a consequence, positive and negative values of one class at the same frequency would disturb or cancel the learning effect. So in all the experimental study we have carried out, image classification is done using only absolute DCT values.

3 Architecture and Visualization

In order to perform the experiment, we have developed a software module based on the Deeplearning4j framework [7]. All images are first transformed into Fourier domain and then the absolute value DCTs are loaded in a three-layer NN consisting of an input layer, a hidden layer with ReLU activation and the softmax output layer. It is possible to connect the image directly to the input layer. So with the software module, we can work both in the spatial or frequency domain. Yet the later option is the focus of our research. After training, a test dataset is used to evaluate the quality of the classification. Our software is capable of calculating the achieved classification accuracy as a value between 0 and 1. An accuracy of 1 (i.e., 100%) means that all images are classified correctly.

In a second module of the software it is possible to visualize the activity of the neurons. After the learning phase, the weights and bias of every neuron are fixed and stored in the trained neural model. During the testing period and for each test image, we calculate the activity of every single neuron in the network. This values are individual for each image and they are stored temporarily. By tracing backwards the neuron activities, i.e. from the last layer (softmax decision layer) over the hidden layer to the input layer, it is possible to determine which parts of the input layer lead to the decision in the last layer. Because the input layer is a 2D array, it is possible to create an activity image. Like the DCT Images they also represent horizontal frequencies in the direction from left to right and vertical frequencies from top to bottom.

4 Experiments

4.1 MNIST dataset

The MNIST dataset [1] consists of 70.000 images of hand written digits, divided into a training set of 60.000 and a test set of 10.000. Each image has the size of 28×28 pixels. The object classes are the digits 0 to 9.

Based on the training set, we train a NN with their absolute value DCTs. On the test set, we store the neuron activities for each test image. By tracing backwards, we are able to produce the activity image for each test image. In order to get an insight about the patterns learned through the NN, we build an average activity image for each object class, using only the correctly classified examples.

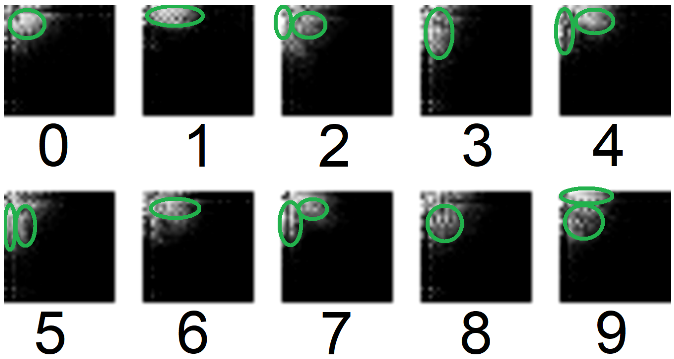

Fig. 2: Average activity images for all object classes on the MNIST dataset

As shown in Fig 2, the spectrum in the activity image of digit 0 is concentrated in the upper left corner, and it consists of frequencies more or less equally in all directions. The digit 1 is a vertical bar in the spatial domain. It is recognized by a dominating horizontal frequency, so in the activity image, it is represented by a bar on top of the image. Digit 2 consists of a horizontal bar at the bottom. This is verified by the concentrated frequency on the left of the upper side of the DCT. The upper part of the digit 2 has again frequencies in all directions. The digit 8 consists of 2 circles, similar to the case of digit 0. So it has frequencies in all directions. Since the two circles of 8 have smaller radius than that of 0, the frequencies of 8 are pushed toward the center on the activity image.

4.2 MNIST displaced dataset

In the following experiment we compared image classification in the spatial domain with that in Fourier Domain: In the MNIST dataset the digits are all centered. Our research is about the benefits of Fourier transform in terms of translation invariance. So we create MNIST displaced dataset by randomly changing the position of the 28 × 28 images in a larger size image of 56 × 56 pixels, as shown in Fig 3. This acts as another test set to be compared with the original MNIST test set.

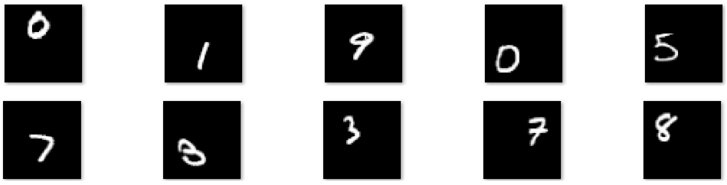

Fig. 3: MNIST displaced dataset.

The results of the comparative study are shown on the left side of Fig 4. On the original test set with digits centered in the input image, classification performance in spatial domain outperforms that of Fourier domain. The benefit of classification in the Fourier domain lies in the fact of translation invariance. So classification with absolute value DCTs outperforms classification on the MNIST displaced dataset, particularly in cases where the hidden layer has a smaller number of neurons.

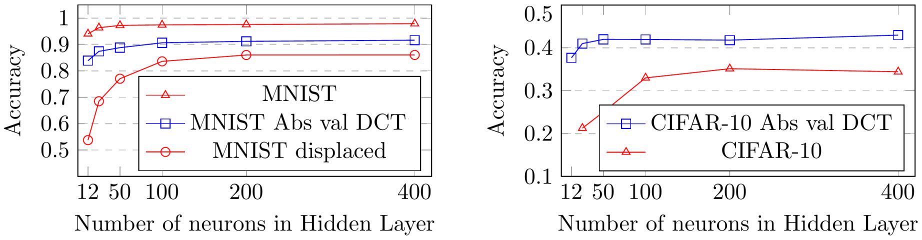

Fig. 4: Comparative Study with different methods and models. Left: Results on the MNIST dataset. Right: Results on the CIFAR-10 dataset.

4.3 CIFAR-10 dataset

The CIFAR-10 dataset [8] consists of 60.000 color images, divided into a training set of 50.000 and a test set of 10.000. The images have the size of 32 × 32 pixels. The ten classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. On this experiment, we first convert all the images into grey-level images so that they can be used together with our neural architecture.

Once a network is trained, we use the same process to create the average activity images, as is shown in Fig. 5.

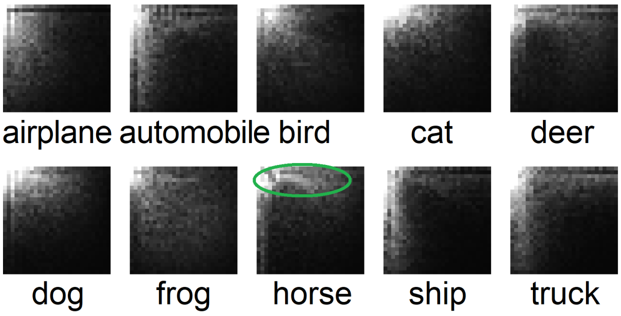

Fig. 5: Average activity images on the CIFAR-10 dataset.

Again we can interpret the activity images: In real images of technical objects like automobiles, trucks and ships, there are horizontal lines, which are vertical frequencies. In the activity images in the frequency domain one can observe white band on the left side of each of these images. Automobiles and trucks are similar to each other, while the later ones have more vertical structures as can be observed in Fig. 5. In the case of frog, it has neither vertical nor horizontal structure. Cats and dogs are also similar. Image of the horse activity has strong vertical structures due to legs. Please note that the images here are only the average activities of correctly classified images grouped over a whole class. The network still has to adapt to different patterns within one class.

We also compared the performance of NNs with different configurations. As shown on the right side of Fig 4, classification result in the frequency domain clearly outperforms that in the spatial domain.

4.4 CIFAR-10 dataset with occlusion

Based on the results shown in Fig. 5, we believe that the strong vertical structures of the legs play an important role in the classification of the horse. In order to verify this, we picked 10 horse pictures, where they have been correctly identified by the NN. Similar to the work of Zeiler et. al. [3], we occlude the upper part of these horse images with a grey rectangle, resulting in images where only legs of the horses are present. The occluded images are shown in Fig. 6. Using the NN trained before, all those half-horse images have been classified correctly, leading to 100% accuracy.

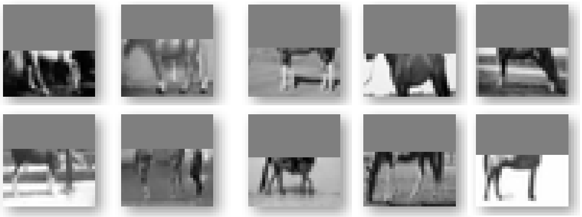

Fig. 6: Partially occluded horse images.

5 Conclusion

By tracing the activity of the neurons and by visualizing them as activity images, it is possible to show the underlying patterns responsible for NN based image classification in Fourier domain. This is useful for gaining more understanding of image classification. Our experiments have shown that in case of architecture with one hidden layer, classification performance in Fourier domain can be superior to its counterpart in the spatial domain. Through obscuring object parts, we have demonstrated that a Fourier domain classifier can recognize horses based on images with only legs visible.

In the future, we will find ways for revealing the different classifiers within one class. It is known that Fourier transform is invariant to translation, but not for rotation and scaling. In order to deal with this problem, we plan to combine the current method with a subsampling layer and perform more extensive study on NN.

References

[1] LeCun, Y., Bouttou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition Proceedings of the IEEE 86(11),pp. 2278-2324 (1998)

[2] Mathieu, M., Henaff, M., LeCun, Y.: Fast Training of Convolutional Networks through FFTs, arXiv: 1312.5851v5 (2014)

[3] Zeiler, M. D., Fergus, R.: Visualizing and understanding convolutional networks. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, ECCV, volume 8689 of Lecture Notes in Computer Science, pp. 818-833. Springer (2014)

[4] Frénay, B., Dumas, B.: Information Visualisation and Machine Learning: Characteristics, Convergence and Perspective, proceedings of the 16th European Symposium on Artificial Neural Networks (ESANN 2016), pp. 623-628., Bruges (2016)

[5] Chen, W., Wilson, J. T., Tyree, S., Weinberger, K. Q. , Chen, Y. : Compressing convolutional neural networks. arXiv:1506.04449 (2015)

[6] Pratt, H., Williams B., Coenen F., Zheng Y.: FCNN: Fourier Convolutional Neural Networks. In: Ceci M., Hollmen J., Todorovski L., Vens C., Dzeroski S. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2017. Lecture Notes in Computer Science, vol 10534. Springer (2017)

[7] Eclipse Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org

[8] Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images, Master thesis, Department of computer Science, University of Toronto (2009)