Recognition of natural texture images

Jaime Enrique Cascante Vega
Universidad de los Andes
je.cascante10@uniandes.edu.co

Angela Castillo Aguirre
Universidad de los Andes
a.castillo13@uniandes.edu.co

1. Introduction

Before deep learning based method became the common approach in the computer vision, textons and SIFT (Scale Invariant Feature Transform) based methods where the state of the art methods [1, 2]. In 2012 A. Krizhevsky, et. al [3] presented Deep convolutional neural networks (CNN) in the context of image classification in the ImageNet dataset. Since then CNN have attracted much attention due to their capacity to recognize thousands of object categories in natural image databases. Their architecture is somehow similar to the structure of the human visual system: both use restricted receptive fields, and a hierarchy of layers which progressively extract more and more abstracted and detailed features [4].

Hubel and Wiesel found that there are specific cells in the visual cortex that respond to specific visual inputs. Determined light orientation or determine input shapes activate specific cluster of neurons. They also observed that as the visual input is *more specific* the cluster size increase. This implies that the organization of the visual system is hierarchical [5, 6]. In [7] the first generation of CNN is introduced for the traditional digit recognition task nowadays widely known from MNIST database. The biological inspiration in CNNs is the one describe before, the first convolutional layers learns local images features like circular edges, different oriented lines, while the deepest layers learns more specific feature of the problem, for example in a face recognition problem the deepest layers will learn eyes, mouth, nose, etc [8]. However the CNN are not like the visual cortex even they are inspirited in it. Last decade due to the increase of the computational power and advent of fast graphical processor unit (GPU) lead to a breakthrough, [3] is the work that used convolutional nets leading to almost halve the error rate for object recognition, which precipitated the rapid adoption of deep learning by the computer vision community.

In this work we use the texture database of the University of Illinois at Urbana-Champaign computer vision group [9]. The database is composed of 25 different classes each class with 40 gray-scale images for a total of 1000 images. Each class correspond to one specific texture, the resolution of each image is 640×480. From top to bottom and left to right the figure below shows a random image for each class bark1, bark2, bark3, wood1, wood2, wood3, water, granite, marble, floor1, floor2, pebbles, wall, brick1, brick2, glass1, glass2, carpet1, carpet2, upholstery, wallpaper, fur, knit, corduroy and plaid.

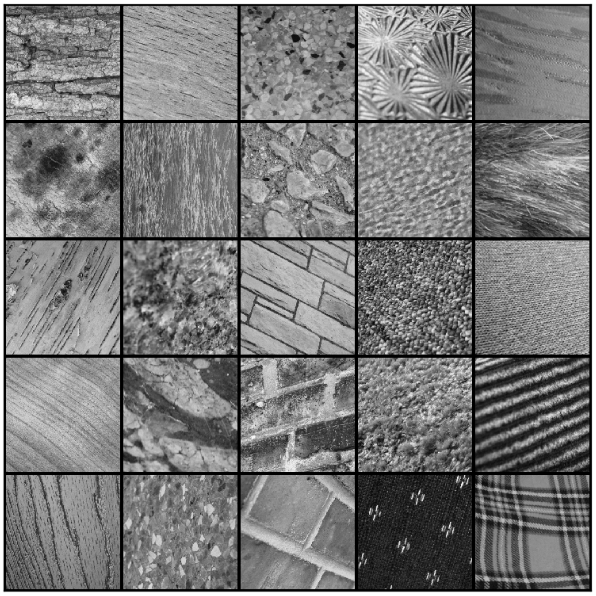

Figure 1: Random image for each texture class.

2. Methodology

With respect to the specific database for this study, there are 20,000 images of 128x128 resolution, of the texture database mentioned above. The training set has 15,000 images, while the validation and test sets have 2500 images each.

2.1. AlexNet

In the present study, some preliminary experiments were carried out before starting with the training phase of the neural network. As a first step, the AlexNet was used. Its architecture consists in the implementation of 5 convolutional layers and 3 fully connected layers, with Relu activation function after each convolutional layer. In this case, the Relu activation function are responsible for adding the non-linearities, instead of the conventional $Tanh()$ function. In addition, a dropout is made before the first and the second fully connected layer. As the resolution of the images of our dataset is 128x128, we had to resize 256 to make use of this network so that the images were not lost by the depth of it.

Regarding the learn rate, it was considered the update of it as it progresses in the epochs, being the initial 0.1 and was multiplied by 0.99 in each iteration, however, this was not efficient, so no significant results are reported from this. Nevertheless, this served to consider a constant learning rate, which was set to 1×10^{-3} for all iterations. The momentum was set at 0.915, which is close to default values. In total, 50 epochs were run, each with a batch size of 10.

2.2. ResNet

AlexNet was the first option to perform the experiments since this network is simple and for some time was the leader in ImageNet. However, the results obtained were not as expected, so we decided to use the ResNet architecture in our data, to check the efficiency of this neural network. Because of this, no major changes were made with respect to the hyperparameters of number of times, learn rate and momentum. However, for this case, it was chosen to increase the batch size to 50 since the computational capacity allowed this increase. An upsampling to 300 was required for this method, so the image would not be lost in the depth of the net.

In general, the ResNet architecture consists in the implementation of different neuronal layers that consider the combination between convolutional layers and Relu activation function. However, the main strategy for this network consists of the so-called ”identity shortcut connection” which basically is jumping from one layer to another, with jumps of one or more layers. For this case, the ResNet18 was used since it is the smallest of the network family for this method.

2.3. Evaluation

To evaluate the performance of the neural network, the loss was calculated using the *CrossEntropyLoss()* function of pytorch, which combines *LogSoftmax()* and *NLLLoss()*. The loss is computed as

$$loss(x, class) = -log\left(\frac{exp(x[class])}{\Sigma_j exp(x[j])}\right)$$

$$loss(x, class) = -(x[class]) + log\left(\Sigma_j exp(x[j])\right).$$

At the end of the epoch, the total average is taken for the accumulated loss of each data in the batch. As the loss is lower, the performance of the neural network for the epoch will be better.

On the other hand, the accuracy was calculated as

$$Acc = \Sigma_j(Predictions = Targets)$$

$$\%Acc = \frac{Acc \times 100}{SetLength}$$

In this way, important information was obtained to determine how well the learning of the network occurred as the batchs pass through it and likewise, the improvement given by the updating of the weights for each epoch was evident.

3. Results and Discussion

According to the first experiment done with AlexNet, no major improvement was found as network training progressed, with an average accuracy of 3% over 15 epochs.

However, when considering the change made to the learn rate parameter, the improvement was much more evident and the result of this experiment is reported in figure 2. In this figure, the relationship between the epochs and the accuracy of the period is reported. As it is possible to see in this figure, as the epochs advanced, a performance improvement is noticed. Nevertheless, the rate of improvement is not very high, although it is a positive derivative. This rate is maintained throughout the ages. Between periods 20 to 30, a considerable improvement is noted, but it is not very substantial to consider this the best training option.

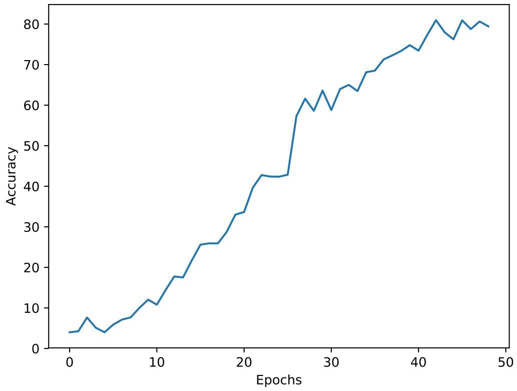

Figure 2: Relationship between accuracy and epochs, in the testing set for the AlexNet network.

Given this, it is considered to train with the proposed network in ResNet. The accuracy results in the train set are presented in figure 3. As it is possible to see here, the first epoch had an accuracy of more than 20% indicating the remarkable improvement of this network, with respect to AlexNet. In addition, this improvement is maintained over approximately the first 10 epochs. Between epochs 10 and 20, what you see is a transition, decreasing the speed of improvement that had been presented. The rate of improvement decreased considerably after epoch 20. In the end, as it can be seen, the neuronal network makes constant improvements; however, at this point, take 3 times for the accuracy to rise 1%. In general, the function that continues this behaviour is constant.

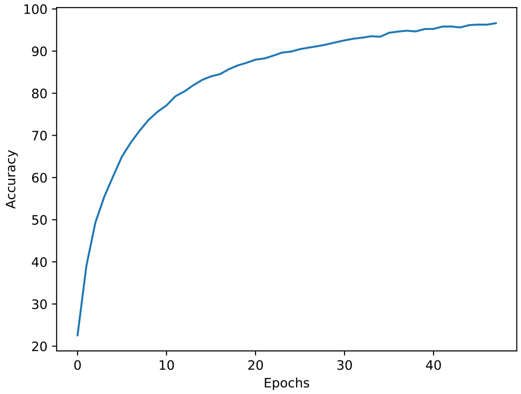

Figure 3: Relationship between accuracy and epochs, in the training set for the ResNet network.

Finally, the result obtained in the test set provided is presented. As it is possible to see in figure 4, the function that is followed is similar and comparable with what is obtained for the train set. The first thing to note is that the accuracy for the first epoch of this test was comparable with that obtained in the train set. In addition, the rate of improvement is high as expected by what was obtained in training, at least for the first 10 epochs. After this moment, the increase in accuracy begins to fluctuate as the epochs progress. In some cases, the improvement is not evident. After epoch 30, there is a specific epoch that is not good, however the network decides to return and look for the best gradient to present convergence.

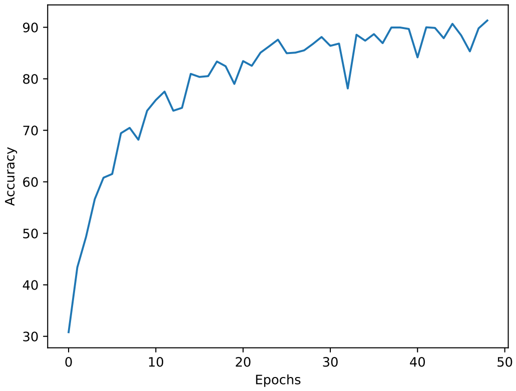

Figure 4: Relationship between accuracy and epochs, in the testing set for the ResNet network.

4. Conclusions

In general, it can be said that convolutional neural networks are methods that work for the classification of images, which in this case are textures. AlexNet is a network that works fairly well compared to some other methods that before deep learning. However, after winning ImageNet, the improvement of AlexNet to ResNet was evident as one of the networks that are still referenced today.

In the case of this study, the improvement shown from AlexNet to ResNet was evident. AlexNet worked well for what was expected, however, the improvement over the epochs is not evident. Considering a prominent improvement as a high speed of improvement as epochs go by. This phenomenon is evident in ResNet, at least during the first epochs, making this an obvious advantage that then dampens the low speed of accuracy increase for the last 15 epochs, approximately.

Finally, it can be said that there are classification strategies, such as convolutional neural networks, that work to differentiate between different categories of a database. In this case, a few hyperparameters were varied from those that can be varied in these networks, so it is believed that the tuning of these parameters could produce better results, at least in this database.

References

[1] T. K. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons.," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[4] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier, “Deep networks can resemble human feed-forward vision in invariant object recognition,” *Scientific Reports*, vol. 6, no. 1, 2016.

[5] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[6] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *The Journal of Physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Comput.*, vol. 1, pp. 541–551, Dec. 1989.

[8] R. Szeliski, *Computer vision*. Springer, 2011.

[9] S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using local affine regions.,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1265–1278, May 2005.