Convolutional Neural Networks for image classification

Nadia Jmour
LA.R.A Laboratory, National Engineering School of Tunis
National Engineering School of Carthage
Tunis, Tunisia
nadiajmour@gmail.com

Sehla Zayen
LA.R.A Laboratory, National Engineering School of Tunis
National Engineering School of Carthage
Tunis, Tunisia
sehla.loussaief@gmail.com

Afef Abdelkrim
LA.R.A Laboratory, National Engineering School of Tunis
National Engineering school of Carthage
Tunis, Tunisia
afef.a.abdelkrim@ieee.org

***Abstract*— This paper describes a learning approach based on training convolutional neural networks (CNN) for a traffic sign classification system. In addition, it presents the preliminary classification results of applying this CNN to learn features and classify RGB-D images task. To determine the appropriate architecture, we explore the transfer learning technique called "fine tuning technique", of reusing layers trained on the ImageNet dataset in order to provide a solution for a four-class classification task of a new set of data.**

***Keywords*— Convolutional neural network, Deep Learning, Transfer Learning, ImageNet.**

I. INTRODUCTION

Image representation for classification task used often feature extraction methods which have been proven to be effective for different visual recognition tasks, [1]. Local binary patterns method is used for texture features extracting. Histograms of oriented gradients are applying for image processing. Usually these types of methods have been used to transform images and describe them for many tasks, [2]. Most of the applied features need to be identified by an expert and then manually coded as per the data type and domain. This process is difficult and expensive in terms of expertise and time.

As a solution, deep learning reduces the task of developing new feature extractor, [3], by automating the phase of extracting and learning features. The proposed traffic sign classification system is able to recognize the traffic sign images put on the road and classify them by exploiting this technology.

There exist many different architectures of deep learning. The model presented in this paper is a classifier system developed by using convolutional neural networks category, [4], which is the most efficient and useful deep neural network used for this type of data, [5]. Therefore, CNNs applied to learn images representation on large-scale datasets for recognition tasks can be exploited by transferring these learning representations on other tasks with limited amount of training data.

To address this problem, we propose using the convolutional neural network AlexNet applied on the large-scale datasets ImageNet, [6] [7], by transferring its learned image representations and reuse them to the classification task with limited training data. The main idea is based on designing a method which reuse a part of training layers of AlexNet.

In the following, problem statement is presented in section II. Sections III introduces the method and the CNN architecture exploited. Initial experiment results using the appropriate CNN architecture which demonstrates that the developed deep neural network achieves a satisfied success rate are described in the first part of section IV. In the second part, the effect of the MiniBatchsize parameter is discussed , [8].

II. PROBLEM STATEMENT

Usually, it is not evident for a driver to keep his eyes everywhere at once while driving. Being concentrated on the road, checking it, looking oncoming traffic, what's behind him, all while trying to control his speed, can become difficult and annoying. To avoid any road accident problem, traffic sign need to be rigid, unique and clear for the driver.

With the Traffic Sign classification system, the risk of warning from a potential hazard ahead can be vastly reduced. Also, with automatic classifying Traffic signs a mandatory problem of self-driving cars can be solved.

The present paper aims to build a classifier system that can determine the type of the traffic sign displayed in an image, and is robust to different real-life conditions such as poor lighting or obstructions by designing an image processing algorithm. As an initial work, four types of traffic sign are used: Nonstop signs, stop sign, green light and red light.

Digit image and face task classification, [9], describe limited variation in appearance and pose. Therefore, these two domains are close to our task and the applied methods

978-1-5386-4449-2/18/$31.00 ©2018 IEEE.

can be efficiently used to traffic sign classification task. Applications and researches on image classification, transfer learning, and deep learning are references on which our method is related and discussed below.

Recent methods of image classification tasks use the bag-of-features pipeline. SIFT descriptors, [10], are using for clustering. Spatial pooling, [11], Histogram encoding, [12] and recent Fisher Vector encoding, [13] are using for feature collection. Although these representations have been given acceptable results, it is not obvious if they are optimal for the task, since it requires a lot of time and effort from experts in the specific domain. This process is difficult and expensive in terms of expertise and time.

Deep learning or deep neural networks reduces the task of developing new feature extractor for every visual recognition problem. This optimization is realized by automating the phase of learning image's representation and using graphics processing units (GPUs), [14], suited to the application's problem.

III. CONVOLUTIONAL NEURAL NETWORK

A. Architecture

Convolutional Networks (ConvNets) are currently the most efficient deep models for classifying images data. Their multi-stage architectures are inspired from the science of biology. Through these models, invariant features are learned hierarchically and automatically, [15]. They first identify low level features and then learn to recognize and combine these features to learn more complicated patterns.

These different levels of features come from different layers of the network. And each layer has specific number of neurons and presented in 3 dimensions: height, width, depth, [16].

To understand convolutional neural network structure, [17], we can observe it as two distinct parts. In input, images are presented as a matrix of pixels. It has 2 dimensions for a grayscale image. The color is represented by a third dimension, of depth 3 to represent the fundamental colors (Red, Green, Blue), [18].

The first part of a CNN is the convolutive part. It functions as a feature extractor of images. In this part, an image is passed through a succession of filters, or convolution kernels, creating new images called convolution maps. Some intermediate filters reduce the resolution of the image by a local maximum operation.

- CONV layer accepting a volume of size [W1×H1×D1] where W1 is the width, H1 is the height and D1 the depth, the outputs of neurons in this type of layers are calculated by applying the product between their weights and a local region they are connected to in the input volume. The obtained output volume [W2×H2×D2] called convolution maps where W2 is the width, H2 is the height and D2 is the depth if we decided to use D2 filters or convolution kernels, [19]. Convolution maps produce a volume equal to [W2×H2×D2] where W2, H2, D2 are given by equations (1), (2), (3):

$$W2 = \frac{W1 - F + 2 * P}{S} + 1 \quad (1)$$

$$H2 = \frac{H1 - F + 2 * P}{S} + 1 \quad (2)$$

$$D2 = K \quad (3)$$

With :

F : spatial extend of the filter.
K : number of filters.
P : zero padding (hyperparameter controlling the output volume).
S : stride (hyperparameter with which we slide the filter).

- RELU layer applying an activation function such as the max(0,x) function, to product elementwise non-linearity. This operation does not affect or change the size of the volume, [20].
- POOL layer inserted between successive Conv layers, applying a downsampling operation along the spatial dimensions width and height. It uses MAX operation to optimize the spatial size of the representation as well as reducing the amount of parameters, [21]. Pool Layer produces a volume [W2×H2×D2] where W2, H2, D2 are given by applying equations (4), (5) and (6) :

$$W2 = \frac{W1 - F}{S} + 1 \quad (4)$$

$$H2 = \frac{H1 - F}{S} + 1 \quad (5)$$

$$D2 = D1 \quad (6)$$

In the end, a feature extractor vector or CNN code concatenate the output informations as a unique vector.

This code is then connected to the input of a second part, consisting of fully connected layers (multilayer perceptron), [22]. The role of this part is to combine the characteristics of the CNN code to classify the image. It determines the class scores, presenting in an output volume of size [1×1×k]. The architecture of this part is a usual multilayer perceptron and

each of the k output neurons or numbers, connecting to all the numbers of the previous layer, correspond to a category of the classification.

B. CNN training

Creating CNN is expensive in terms of expertise, equipment and the amount of needed data. The first step is to fix the architecture by fixing the number of chosen layers, their sizes and matrix operations that connect them, [23]. The training consists then, of optimizing the network's coefficients to minimize the output classification error.

This training can take several weeks for the best CNNs, with many GPUs working on hundreds of thousands of annotated images. Research teams specialized in improving CNN publish their technical innovations, so the complexity of creating CNN can be avoided by adapting publicly available pre-trained networks. These techniques are called transfer learning, [24], which consist on transferring knowledge from the related source to the target domain. These pre-trained neural network can be used in two ways:

- Automatic feature extractor of images : exploits only the convolutive part of a pretrained network. It uses it as an automatic feature extractor of images, to feed the classifier of our choice. It keep only the convolutive part. This part is called frozen, to express the absence of training. This network takes an image in good format and outputs the CNN code. Each image in the dataset is thus transformed into a feature vector, which is used to drive a new classifier, [25]. This method has many practical interests:

  The image is transformed into a small vector, which extracts features that are usually very relevant. This reduces the size of the problem, [26].

  Feature extraction is being performed only once per image, it can be performed quickly on CPU. The machine learning libraries are usually sequential and also run on CPUs.

  This method makes it possible to exploit the power of the CNNs without investing in GPUs, [27].

- Fine tuning : an initialization of the target model, which is then retrained more finely to deal with the new classification problem, [28]. Here we use an architecture carefully optimized by specialists, and we take advantage of features extraction capabilities learned on a large quality dataset. Fine Tuning on images consists of some sort of taking a visual system already well trained on a classification task to refine it on a similar task. The only necessary change to the network is the adaptation of the last layer, [29]. For training, it is possible to freeze the initial layers of the neural network, and to adapt only the final layers for the new classification problem. Freezing all convolutional layers corresponds to the first method presented, with final classifier a pre-initialized multilayer perceptron.

To learn features for our traffic sign classification task, we apply ConvNets combined with fine tuning technique. We used the pre-trained convolutional neural network AlexNet which is trained for 1000 possible categories on the large dataset ImageNet in the Large Scale Visual Recognition Challenge (ILSVRC-2012), [30], containing over then 1.2 million images. Results were remarkable and achieved a top-5 error of 15.3%.

IV. Experiments

In this section, we first detailed the architecture of the proposed model. Next, we present experimental results of our method based on transfer learning technique for the traffic sign classification task dataset. Finally, we discuss the effect of one of the hyperparameters of a deep neural network in our model.

A. Dataset

The traffic sign dataset contains more than 360 images in total, divided into different classes. To avoid using the testing data, we leave 180 images from the training set for validation and 180 test images featuring among four classes "stop sign", "non stop sign", "Green light" and "Red light". Both training and testing data are distributed over these categories.

B. CNN developped architecture

Adapted network exploited in our method is AlexNet deep neural network. AlexNet was among the first well performed convolutional neural network in the computer vision community. This CNN has shown successful results while training on difficult ImageNet dataset. The training of the model is released on two GTX 580 GPUs for five to six days, [31], using batch stochastic gradient descent algorithm.

The network was made up of 5 internal convolutional layers: C1, C2, C3, C4, C5, pooling layers, dropout layers, and 3 fully connected layers : FC6, FC7, FC8. It was used for classification with 1000 possible categories, [32].

The input of the architecture takes images of size [227×227×3] with a zero padding P=0. On the first Convolutional Layer, AlexNet applied 96 convolution kernels with size of F=11 by striding it among the input volume with a strider S=4. The output volume had for size [55×55×96] where height and width are W= H= 55 = (227-11)/4+1 and depth K=96. The total number of neurons in this layer is 55×55×96=290400 neurons.

Each of the 290400 neurons was connected to a local region of [11×11×3] in the input, and all of the 96 neurons are connected with different values of weights to the same region of size [11×11×3] in the input volume. The rest of successive layers and filters applied are presented on fig.1.

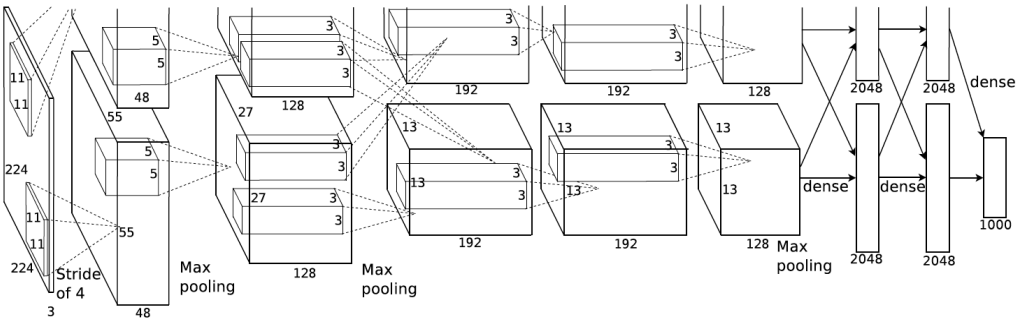

Fig. 1. AlexNet architecture, [33].

C. *Setting and results*

To realize our classification task and evaluate results of the related method on this target task, we exploit AlexNet pretrained on 1000 categories by transferring its training and optimizing parameters to the traffic sign classification task. The model is trained with stochastic gradient descent algorithm and we apply 4 training experiments followed by 3, 4, 5, 6 epochs with learning rate 0.0001 and a MiniBatchsize 30 which took an importing time on Intel® HD Graphics 3000 CPUs. Results of this experiment are reported in Table 1.

TABLE 1. Classification results.

	Number of iterations	**Training time (s)**	**Accuracy (%)**
3 Epochs	18	11668.74	0.6592
4 Epochs	24	16125.34	0.8380
5 Epochs	30	26145.18	0.8444
6 Epochs	36	29337.20	0.8620

Classification rates were obtained for different number of epochs. We observe the increase in the value of the classification rate when incrementing the number of Epochs. We managed to obtain a maximum accuracy of 0.8620 for 6 training epochs by using the optimized architecture AlexNet and taking advantage of the features extraction capabilities learned on ImageNet dataset. Fig. 2 shows an exemple of test classification for some labels.

Fig. 2. Test classification for some labels.

D. *MiniBatchsize effect*

Several parameters of the training network have an influence on the obtained results, such as the dataset's size, the initial learning rate and the number of layers. In this part, the influence of the MiniBatchsize training parameter is discussed.

MiniBatchsize or Batch training consists of backpropagating the error of classification by groups of images, [34]. To observe the effect of this parameter we propose to train the CNN for different values of MiniBatchsize. The obtained results for the values of 10 and 60 are presented in tables 2 and 3.

TABLE 2. Training results for Minbatchsize 60.

	Number of Iterations	**Training time (s)**	**Accuracy (%)**
3 Epochs	9	12091.69	0.5866
4 Epochs	12	17611.06	0.6089
5 Epochs	15	39255.54	0.6816
6 Epochs	18	42480.20	0.7541

Fig. 3. Test classification for some labels with minibatchsize 60.

The results of training for different numbers of epochs and a large Minibatchsize 60, gave classification rates during a significant training time varying between 0.5866 and 0.7541. Comparing to results obtained in table 1, the decrease of the rate explain the problem of memorization shown in Fig. 3 where the « Don't stop sign » has been misclassified and considered as a « Stop Signs » category.

Table 3 shows that the training for different number of epochs gives important values of the classification rate. These values vary between 0.8429 and 0.9333 and are calculated during a training time of 25882.40s. This low value of MiniBatch makes it possible to perform the classification task with more precision.

TABLE 3. Training results for MiniBatchsize 10.

	Number of Iterations	**Training time (s)**	**Accuracy (%)**
3 Epochs	54	11589.23	0.8429
4 Epochs	72	15173.47	0.8939
5 Epochs	90	22263.47	0.8944
6 Epochs	108	25882.40	0.9333

E. Discussion

The combination of AlexNet and fine tuning technique consisting on taking a visual system already well trained on a classification task to refine it on a similar task, has enabled the use of a carefully optimized architecture by specialists, and take advantage of features extraction capabilities learned on a large quality dataset.

The only necessary change to the network is to adapt the last layer. Our problem has 4 categories, while the initial training of the AlexNet was done on 1000 categories. It was therefore necessary to change the last layer of AlexNet and replacing it by a layer of four neurons.

The proposed method consists on transferring trained weights of layers in the first part of the adapted and pretrained network C1, C2, C3, C4, C5, FC6 and FC7. Then we implement the last adapted layer FCL8_2. The first part is freezing, and results were obtained by applying training only in the second part from FC6 layer to the FCL8_2 adapted layer. By this way the convolutional neural network will be better adapted with our classification task. This strategy has shown remarkable performance and optimized training time.

Then, we have analyzed the effect of the minibatchsize. After fine-tuning training iterations this model scored 58.66% accuracy with an important minibatchsize 60. Then, the model scored 93.33% accuracy on the test set with a Minibatchsize 10 which is not too bad. Using this parameter is faster than calculating the error over the entire training data at each iteration. It is more stable than working image by image, because the error gradients have less variance. Too many images per batch can cause memory problems when running the code.

V. CONCLUSION

For feature images extraction and learning, deep neural networks are very effective but these systems unfortunately takes a long time for training layers with simple hardware.

In this paper, we have presented a method for learning traffic sign images for classifier system. This application is released using deep learning to build this classifier, by training on 360 images. The implementation of this type of learning, particularly the convolutional neural network for the classification of image data, and by exploiting AlexNet combined with the technique of transfer learning constitute the objectif of our work. We have demonstrated that fine tuning parameters is a very important and useful technique in the training. The application of this fine tuning has given a reasonable results and has effect on the time of training.

Our experimental results demonstrate also the effect of the MiniBatchsize parameter. The adjustment of this parameter is important in the training process, the results and the training time depend on the chosen values. Too many batch images can cause memory problems when running the code. Time of training was important because of the use of a simple hardware computer.

Deep learning is worth understanding and it is practical efficient since the training time is controlled. In future research, our system will be ameliorated with the use of genetic algorithms which will be added to optimize and solve the classification feature extraction problem in order to optimize the number of parameters in this task.

References

[1] Redmon J, and Angelova A, "Real-time grasp detection using convolutional neural networks", IEEE International Conference on Robotics and Automation, pp. 1316–1322, 2015.

[2] Hang Chang, Cheng Zhong, Ju Han, Jian-Hua Mao, "Unsupervised Transfer Learning via Multi-Scale Convolutional Sparse Coding for Biomedical Application." IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 janvier 2017.

[3] Zhou, X., Yu, K., Zhang, T., & Huang, T. "Image classification using super-vector coding of local image descriptors." In ECCV,2010.

[4] van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M, "Evaluating color descriptors for object and scene recognition", IEEE Transactions on Pattern Analysisand Machine Intelligence." 1582– 1596. 2010.

[5] Howard, A. , "Some improvements on deep convolutional neural network based image classification." ICLR, 2014.

[6] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., "ImageNet: A large-scale hierarchical image database." In CVPR, 2009.

[7] Ahonen, T., Hadid, A., and Pietikinen, "M. Face description with local binary patterns: Application to face recognition." Pattern Analysis and Machine Intelligence, 2037–2041. 2016.

[8] K. Hornik, M. Stinchcombe, H. White, "Multilayer Feedforword Networks are Universal Approximators." Neural Networks, pp. 359-366, 1989.

[9] G. Cybenko, "Approximation by superpositions of a sigmoidal function." Math. Contr. Signals Syst., pp. 303-314, 1989.

[10] P.Sermanet, D.Eigen, X.Zhang, M.Mathieu, R.Fergus,and Y.LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv:1312.6229, 2013.

[11] H. B. Burke, "Artificial neural networks for cancer research: Outcome prediction." Sem. Surg. Oncol, vol. 10, pp. 73–79, 1994.

[12] H.B. Burke, P.H. Goodman, D.B. Rosen, D.E. Henson, J.N. Weinstein, F.E. Harrell, J.R. Marks, D.P. Winchester, D.G. Bostwick, "Artificial neural networks improve the accuracy of cancer survival prediction." Cancer, vol. 79, pp. 857-862, 1997.

[13] J. Lampinen, S. Smolander, and M.Korhonen, "Wood surface inspection system based on generic visual features." Industrial Applications of Neural Networks, F. F. Soulie and P. Gallinari, Eds, Singapore: World Scientific, pp. 35-42, 1998.

[14] T. Petsche, A. Marcantonio, C. Darken, S. J. Hanson, G. M. Huhn, I. Santoso, "An autoassociator for on-line motor monitoring.", Industrial Applications of Neural Networks, F. F. Soulie and P. Gallinari, Eds, Singapore: World Scientific, pp. 91-97, 1998.

[15] A. Sifaoui, A. Abdelkrim, M. Benrejeb, "On RBF neural network classifier design for iris plants", The 37th International Conference on Computers and Industrial Engineering, pp.113-118, Alexandrie, Octobre 2007.

[16] Sinno Jialin Pan and Qiang Yang, Fellow. "A Survey on Transfer Learning." IEEE Transactions on knowledge and data engineering, Vol. 22, No. 10, October 2010.

[17] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. "Blocks that shout: Distinctive parts for scene classification." CVPR, 2013.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." CVPR, 2014.

[19] 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 Computation, 1(4):541–551, 1989.

[20] Y.Boureau, F. Bach, Y. LeCun, andJ. Ponce. "Learning midlevel features for recognition." CVPR, 2010.

[21] Marc Parizeau. "Le perceptron multicouche et son algorithme de rétropropagation de l'erreur." Département de génie électrique et de génie informatique, Université Laval, 10 Septembre 2014.

[22] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing. "Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks." ECCV,2008.

[23] D. Lowe. "Distinctive image features from scale-invariant keypoints." IJCV, 60(2):91–110,2004.

[24] Y. LeCun, L. Bottou, and J. HuangFu. "Learning methods for generic object recognition with invariance to pose and lighting." CVPR, 2004

[25] F. Perronnin, J. S´anchez, and T. Mensink. "Improving the fisher kernel for large-scale image classification." ECCV, 2010.

[26] : P.Sermanet,D.Eigen,X.Zhang,M.Mathieu,R.Fergus,and Y.LeCun. "Overfeat: Integrated recognition, localizationand detection using convolutional networks." arXiv:1312.6229, 2013.

[27] : Schmidhuber, J. "Multi-column deep neural networks for image classification." CVPR. 2012.

[28] M. A. Ranzato, C. Poultney, S. Chopra, and Y. Lecun. "Efficient learning of sparse representations with an energy-based model." Advances in Neural Information Processing Systems (NIPS), 2006.

[29] Y. LeCun, F.-J. Huang, and L. Bottou. "Learning methods for generic object recognition with invariance to pose and lighting." Computer Vision and Pattern Recognition, 2004.

[30] S. Behnke. "Hierarchical Neural Networks for Image Interpretation." Lecture Notes in Computer Science. Springer, 2003.

[31] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. "Greedy layer-wise training of deep networks." Neural Information Processing Systems, 2007