Comparación entre el descriptor de características HOG usando SVM y Redes Neuronales Convolucionales usando SVM para Clasificación de Imágenes

Comparison of HOG Feature Descriptor using SVM with Convolutional Neural Network using SVM for Image Classification

Herwin Alayn Huillcen Baca[1], Flor de Luz Palomino Valdivia[2], Ivan Soria Solis[3]
[1] Professional School of Systems Engineering, José María Arguedas National University, Peru. hhuillcen@unajma.edu.pe, 0000-0001-9385-7940
[2]Professional School of Systems Engineering, José María Arguedas National University, Peru. fpalomino@unajma.edu.pe, 0000-0003-4638-5762
[3]Professional School of Systems Engineering, José María Arguedas National University, Peru. isoria@unajma.edu.pe, 0000-0003-4202-9402

Abstract
Computer vision applications are present in almost every activity in the world, the classification of images is one of them, however, it still presents some problems in the accuracy of its classifications, which can cause problems when these applications are used in systems that require a lot of certainty of classification. Many innovations, methods, and techniques have been proposed so far, they highlight with greater success the use of Convolutional Neural Networks (CNN), Support Vector Machine (SVM), selectors and feature descriptors.
In this work, we emulate an architecture that combines a convolutional neural network (CNN) with a linear SVM for image classification and other method based on histogram of orient gradient (HOG) feature descriptor using SVM, both works were compared to assess the accuracy, taking as dataset the MNIST hand-written digit dataset the Fashion-MNIST dataset.
The architecture that combines a convolutional neural network (CNN) with a linear SVM for image classification got better accuracy than the method HOG feature descriptor using SVM, with values of 99.32% and 99.10% respectively, tested with MNIST hand-written digit dataset and values of 99.29% and 87.29% respectively, tested with Fashion-MNIST dataset.

Keywords: [Convolutional Neural Networks], [Support Vector Machine], [HOG feature descriptor] , [image classification] , [MNIST] , [Fashion MNIST] , [HOG feature descriptor].

1. Introduction

One of the most popular applications of computer vision is the image classification, in which given a set of images that are labeled with a category, we are asked to predict these categories for a new set of test images and measure the presicion of the predictions, this precision is very important, e.g. if it were used in vehicle license plate detection to report them as stolen, a one-digit classification error could lead to a big problem.
There are many methods for image classification, including multiclass Support Vector Machine (SVM) (CORTES, CORINNA & VAPNIK, 1995), Gradient Oriented Histogram (HOG) (Dalal & Triggs, 2010), as an efficient feature descriptor, which combined with a classification algorithm shows excellent results, Convolutional Neural Networks (CNN) (Khan et al., 2020), as the method most used in recent research.
In this paper, we emulate the architecture proposed by Agarap, A. F. (Agarap, 2017), which combines a convolutional neural network (CNN) and a linear SVM for image classification, the CNN employed in this study is a simple 2-Convolutional Layer with Max Pooling model, this work is based on the work of Tang, Y. (Tang, 2013), Deep learning using linear support vector machines. Also, we emulate the classification based on HOG feature descriptor using SVM, proposed by Greeshma, K. & Sreekumar (Greeshma & Sreekumar, 2019).

Both works were tested on the MNIST hand-written digit dataset (Yann LeCun, Corinna Cortes, 2010), and the Fashion-MNIST dataset (Xiao et al., 2017), Finally we make a comparison of classification accuracy between the two works, the architecture proposed by Agarap, A. F. (Agarap, 2017), got better accuracy than the method HOG feature descriptor using SVM, proposed by Greeshma, K. & Sreekumar (Greeshma & Sreekumar, 2019).

2. Background

2.1. Support Vector Machine (SVM)

The support vector machine (SVM) was proposed by Vapnik, V. (CORTES, CORINNA & VAPNIK, 1995), for classification in supervised learning. The objective is to calculate the optimal hyperplane of separation, in (Eq. 1.).

$$f(w,x) = w \cdot x + b \quad (1)$$

and to separate in two classes the dataset, with features $x \in \mathbb{R}^2$. SVM learns the parameters w by solving an optimization problem in (Eq. 2).

$$min\frac{1}{p}w^T w + C\sum_{i=1}^{p} max(0,1y_i'(w^T x_i + b)) \quad (2)$$

where w^T is the L1 norm), C is the penalty parameter, y_i is the actual label, and $(w^T x_i + b)$ is the predictor function. Eq. 2 is called L1-SVM, with the standard hinge loss. With the standard hinge loss. Its differentiable counterpart, L2-SVM (Eq. 3), provides more eficients results [2].

$$min\frac{1}{p} \| w \|_2^2 \, w + C\sum_{i=1}^{p} max(0,1y_i'(w^T x_i + b))^2 \quad (3)$$

where $\| w \|_2$ is the Euclidean norm (also known as L2 norm), with the squared hinge loss.

2.2. Convolutional Neuronal Network (CNN)

(Khan et al., 2020), describe a Convolutional Neural Networks (CNN) that are a special type of Neural Networks, which have shown exemplary performance on several competitions related to Computer Vision and Image Processing. Interesting application areas of CNN include Image Classification and Segmentation, Object Detection, Video Processing, Natural Language Processing, Speech Recognition, etc. The powerful learning ability of deep CNN is largely due to the use of multiple feature extraction stages that can automatically learn representations from the data. Availability of a large amount of data and improvements in the hardware technology have accelerated the research in CNNs, and recently very interesting deep CNN architectures have been reported.

In fact, several interesting ideas to bring advancements in CNNs have been explored such as the use of different activation and loss functions, parameter optimization, regularization, and architectural innovations. However, the major improvement in representational capacity of the deep CNN is achieved through architectural innovations. Especially, the idea of exploiting spatial and channel information, depth and width of architecture, and multi-path information processing has gained substantial attention. Similarly, the idea of using a block of layers as a structural unit is also gaining popularity. This survey thus focuses on the intrinsic taxonomy present in the recently reported deep CNN architectures and consequently, classifies the recent innovations in CNN architectures into seven different categories. These seven categories are based on spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention. Additionally, the elementary understanding of CNN components, current challenges and applications of CNN are also provided.

2.3. Histogram of Oriented Gradients (HOG)

HOG was first introduced by (Dalal & Triggs, 2010), for the human detection and it is one of the most popular and successful feature descriptors in pattern recognition and computer vision.

One of the simple and effective feature extraction methods is HOG feature descriptor. It is a fast and efficient feature descriptor in compare to the SIFT and LBP due to the simple computations, it has been also shown that HOG features are successful descriptor for detection. Mainly it is used for object detection in image processing and computer vision.

3. Related Work

(Marée et al., 2004), they evaluated seven machine learning algorithms for image classification including their recent approach that combines building of ensembles of extremely randomized trees and extraction of sub-windows from the original images. For the approach to be generic, all these methods are applied directly on pixel values without any feature extraction. They compared them on four publicly available datasets corresponding to representative applications of image classification problems: handwritten digits (MNIST), faces (ORL), 3D objects (COIL-IOO), and textures (OUTEX). A comparison with studies from the computer vision literature shows that generic methods can come remarkably close to specialized methods.

In the paper of Palvanov (Palvanov & Cho, 2018), they implemented four models on the basis of unlike algorithms which are capsule network, deep residual learning model, convolutional neural network and multinomial logistic regression to recognize handwritten digits. These models have unlike structure and they have showed a great results on MNIST before so we aim to compare them in real-time environment. The dataset MNIST seems most suitable for this work since it is popular in the field and basically used in many state-of-the-art algorithms beyond those models mentioned above. They purpose revealing most suitable algorithm to recognize handwritten digits in real-time environment. Also, they give comparisons of train and evaluation time, memory usage and other essential indexes of all four models.

(Perez & Wang, 2017), In this paper, they explore and compare multiple solutions to the problem of data augmentation in image classification. Previous work has demonstrated the effectiveness of data augmentation through simple techniques, such as cropping, rotating, and flipping input images. They artificially constrain their access to data to a small subset of the ImageNet dataset, and compare each data augmentation technique in turn. One of the more successful data augmentations strategies is the traditional transformations mentioned above. They also experiment with GANs to generate images of different styles. Finally, they propose a method to allow a neural net to learn augmentations that best improve the classifier, which we call neural augmentation. They discuss the successes and shortcomings of this method on various datasets.

4. Methods

4.1. HOG Feature Descriptor Using SVM

We implement a program that replicates HOG feature descriptor using SVM, proposed by (Greeshma & Sreekumar, 2019), programming in Python and we use the OpenCV library from Python, the figure "Fig. 1" show the pipeline of this proposal:

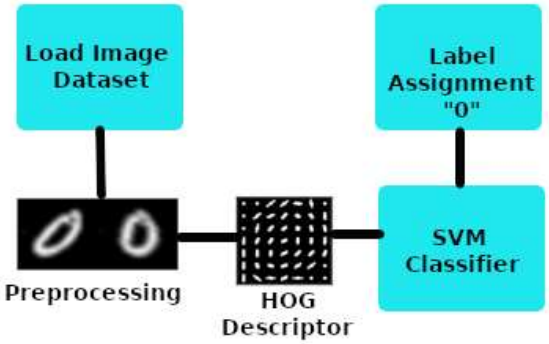


Fig. 1: Pipeline of the propodal of HOG feature descriptor using SVM
Source: Own elaboration

The proposal load the dataset of the MNIST hand-written digit dataset (Yann LeCun, Corinna Cortes, 2010) and the Fashion-MNIST dataset (Xiao et al., 2017), in the case of MNIST dataset, a Deskewing Preprocessing was done, aligning digits before building a classifier similarly produces superior results. In the case of handwritten digits, we do not have features for alignment. However, an obvious variation in writing among people is the slant of their writing. Some writers have a right or forward slant where the digits are slanted forward, some have a backward or left slant, and some have no slant at all. We can help the algorithm quite a bit by fixing this vertical slant so it does not have to learn this variation of the digits. The figure "Fig. 2" show the original digit in the first column and it's deskewed in second colum.

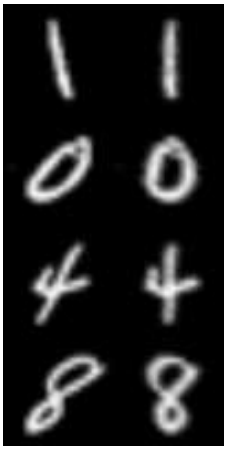


Fig. 2: Deskewing Preprocessing
Source: Own elaboration

This deskewing of simple grayscale images can be achieved using image moments. OpenCV has an implementation of moments and it comes in handy while calculating useful information like centroid, area, skewness of simple images with black backgrounds.
For the feature extraction process using HOG, the cell size used was 14x14, although the proposal indicates to use a 4x4 cell size, but it did not give us good results in accuracy. The next step is to create a histogram of gradients in these 14×14 cells size. The histogram contains 9 bins corresponding to angles 0, 20, 40 … 160, looking the magnitude and direction of the gradient of the same 14×14 cell-size. A bin is selected based on the direction, and the vote (the value that goes into the bin) is selected based on the magnitude. This vector is normalized and scanning the other cells of the image has the feature vector resulting in size 81x1.
Both training and test images are transformed into descriptors using HOG, then an SVM classifier with RBF Gaussian kernel is trained and with a regularization parameter of C=1..

4.2. HOG Feature Descriptor Using SVM

We also implement the architecture proposed by Agarap (Agarap, 2017), which combines a convolutional neural network (CNN) and a linear SVM for image classification. The convolutional layer consists of a filter, for instance, 5×5x1. The convolutian layer is used to "slide" through the width and height of an input image, and compute the dot product of the input's region and the weight learning parameters. This in turn will produce a 2-dimensional activation map that consists of responses of the filter at given regions.
Consequently, the pooling layer reduces the size of input images as per the results of a convolutional filter. Next, the number of parameters within the model is also reduced called down-sampling. mappings. The commonly-used activation function these days is the ReLU function. ReLU is commonly-used over tanh and sigmoid for it was found out that it greatly accelerates the convergence of stochastic gradient descent compared the other two functions.
Furthermore, compared to the extensive computation required by tanh and sigmoid, ReLU is implemented by simply thresholding matrix values at zero.
We replicate the proposal CNN architecture with the following layers:

- (1) INPUT: 28 x 28 x 1
- (2) CONV5: 5 x 5 size, 32 filters, 1 stride
- (3) ReLU: max(θ, h θ (x))
- (4) POOL: 2 x 2 size, 1 stride
- (5) CONV5: 5 x 5 size, 64 filters, 1 stride
- (6) ReLU: max(θ, h θ(x))
- (7) POOL: 2 x 2 size, 1 stride
- (8) FC: 1024 Hidden Neurons
- (9) DROPOUT: p = 0.5
- (10) FC: 10 Output Classes

At the 10th layer of the CNN, instead of the conventional

softmax function with the cross entropy function (for computing loss), the L2-SVM is implemented. That is, the output shall be translated to the following case y∈{+1,-1}, and the loss is computed by (Eq. 3). The weight parameters are then learned using Adam Optimizer [14].
the hyper-parameters used for CNN-SVM model was:

- Batch Size: 128
- Dropout Rate: 0.5
- Learning Rate: 1e-3
- Steps: 1000
- SVM-C: 1

5. Experiment ans Results

5.1. Used dataset

For both methods, we use the dataset of the MNIST hand-written digit dataset (Yann LeCun, Corinna Cortes, 2010) and the Fashion-MNIST dataset (Xiao et al., 2017), the reason for their choice is that MNIST is widely used in image classification research, however, there are researchers, who claim that this dataset does not represent complexity for the current state of the art and they suggest using the Fashion-MNIST dataset, in this way we use both for the experiments. The datasets have 60,000 grayscale images of 28x28, with 10 classes, we will use 50,000 images for training and 10,000 for testing. See “Fig. 4” and “Fig. 5” for un example of both dataset.

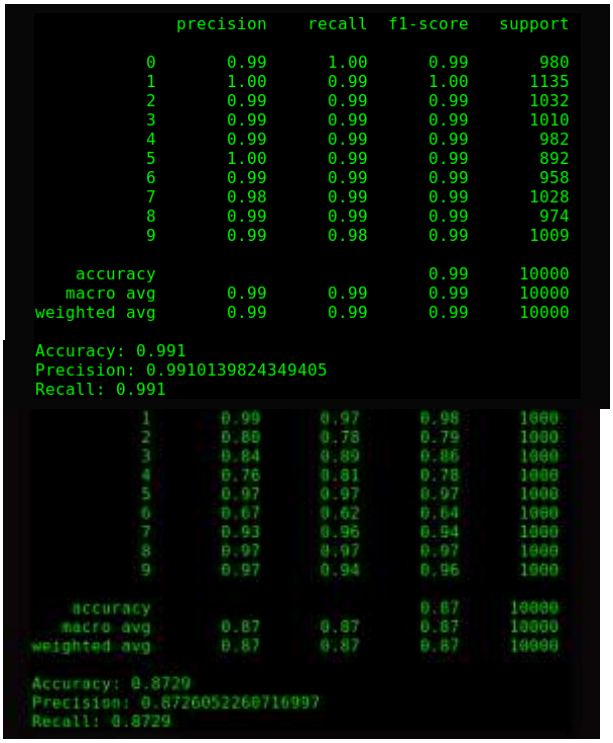
Fig. 4: MNIST hand-written digit dataset
Source: Own elaboration

Fig. 5: Fashion-MNIST dataset
Source: Own elaboration

5.2. Result of HOG Feature Descriptor Using SVM

The figure “Fig. 6” and “Fig. 7”, show the results using dataset Fashion-MNIST and MNIST respectivelly, for each class and the average:

Fig. 6: General Result using dataset Fashion-MNIST with method HOG-SVM
Source: Own elaboration



	precision	recall	f1-score	support
0	0.99	1.00	0.99	980
1	1.00	0.99	1.00	1135
2	0.99	0.99	0.99	1032
3	0.99	0.99	0.99	1010
4	0.99	0.99	0.99	982
5	1.00	0.99	0.99	892
6	0.99	0.99	0.99	958
7	0.98	0.99	0.99	1028
8	0.99	0.99	0.99	974
9	0.99	0.98	0.99	1009
accuracy			0.99	10000
macro avg	0.99	0.99	0.99	10000
weighted avg	0.99	0.99	0.99	10000

Accuracy: 0.991
Precision: 0.9910139824349405
Recall: 0.991

1	0.99	0.97	0.98	1000
2	0.80	0.78	0.79	1000
3	0.84	0.89	0.86	1000
4	0.76	0.81	0.78	1000
5	0.97	0.97	0.97	1000
6	0.67	0.62	0.64	1000
7	0.93	0.96	0.94	1000
8	0.97	0.97	0.97	1000
9	0.97	0.94	0.96	1000
accuracy			0.87	10000
macro avg	0.87	0.87	0.87	10000
weighted avg	0.87	0.87	0.87	10000

Accuracy: 0.8729
Precision: 0.8726052260716997
Recall: 0.8729

Fig. 7: General results using dataset MNIST with method HOG-SVM
Source: Own elaboration

It is observed that the accuracy value in the MNIST and Fashion-MINIST dataset is 99.10% and 87.29% respectively. In this case this method gave better results with the MNIST dataset.

5.3. Result of Convolutional Neural Network Using SVM

The figure “Fig. 8” and “Fig. 9”, show the results of accuracy and training loss respectivelly, using dataset Fashion-MNIST, after 10000 steps.

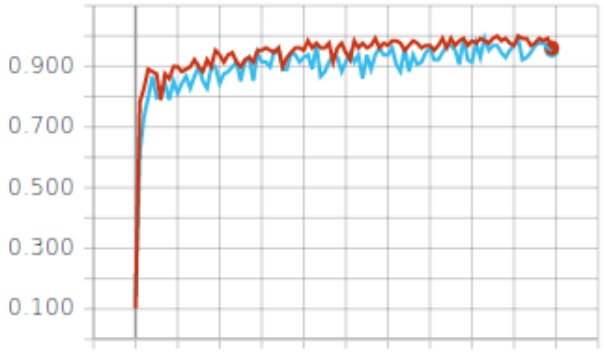


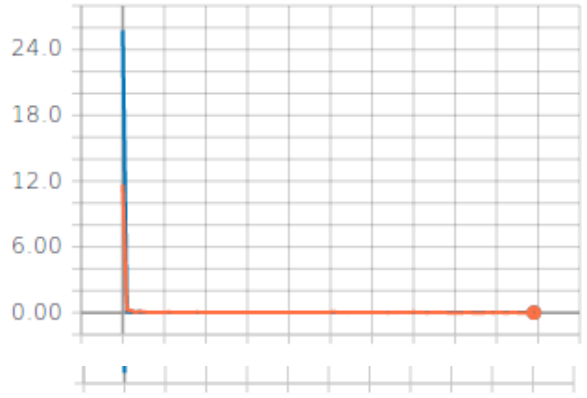
Figure 8: Results of accucary using dataset Fashion-MNIST, after 10000 steps with method CNN-SVM
Source: Own elaboration

Figure 9: Results of training loss using dataset Fashion-MNIST, after 10000 steps with method CNN-SVM
Source: Own elaboration

The figure “Fig. 10” and “Fig. 11”, show the results of accuracy and training loss respectivelly, using dataset MNIST, after 10000 steps.

Figure 10: Results of accucary using dataset MNIST, after 10000 steps, using method CNN-SVM
Source: Own elaboration

Figure 11: Results of training loss using dataset MNIST, after 10000 steps, using method CNN-SVM
Source: Own elaboration



As result, the accuracy value in the MNIST and Fashion-MINIST dataset is 99.32% and 99.29% respectively. Also this method gave better results with the MNIST dataset.
Finally, table 1 shows the final result of both methods with the two datasets.

Table 1: Result of accuracy for both methods with the two datasets

Method/Dataset	HOG Feature Descriptor Using SVM	Convolutional Neural Network Using SVM
MNIST	99.10%	99.32%
Fashion-MNIST	87.29%	99.29%

Source: Own elaboration

6. Conclusions

The architecture proposed by (Agarap, 2017), which combines a convolutional neural network (CNN) and a linear SVM for image classification got better accuracy than the method HOG feature descriptor using SVM, proposed by (Greeshma & Sreekumar, 2019), with values of 99.32% and 99.10% respectively, tested with MNIST hand-written digit dataset (Yann LeCun, Corinna Cortes, 2010).
Similarly, The architecture proposed by (Agarap, 2017), which combines a convolutional neural network (CNN) and a linear SVM for image classification got better accuracy than the method HOG feature descriptor using SVM, proposed by (Greeshma & Sreekumar, 2019), with values of 99.29% and 87.29% respectively, tested with Fashion-MNIST dataset (Xiao et al., 2017).

Referencias

Agarap, A. F. (2017). *An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification*. 5–8. http://arxiv.org/abs/1712.03541

CORTES, CORINNA & VAPNIK, V. (1995). Support-Vector Networks. *Machine Learning*, 273-297.

Dalal, N., & Triggs, B. (2010). Histograms of Oriented Gradients for Human Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 886–893.

Greeshma, K. V., & Sreekumar, K. (2019). Fashion-MNIST classification based on HOG feature descriptor using SVM. *International Journal of Innovative Technology and Exploring Engineering*, *8*(5), 960–962.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 1–70. https://doi.org/10.1007/s10462-020-09825-6

Marée, R., Geurts, P., Visimberga, G., Piater, J., & Wehenkel, L. (2004). A Comparison of Generic Machine Learning Algorithms for Image Classification. *Research and Development in Intelligent Systems XX*, *January 2014*, 169–182. https://doi.org/10.1007/978-0-85729-412-8_13

Palvanov, A., & Cho, Y. I. (2018). Comparisons of deep learning algorithms for MNIST in real-time environment. *International Journal of Fuzzy Logic and Intelligent Systems*, *18*(2), 126–134. https://doi.org/10.5391/IJFIS.2018.18.2.126

Perez, L., & Wang, J. (2017). *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. http://arxiv.org/abs/1712.04621

Tang, Y. (2013). *Deep Learning using Linear Support Vector Machines*. http://arxiv.org/abs/1306.0239

Xiao, H., Rasul, K., & Vollgraf, R. (2017). *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 1–6. http://arxiv.org/abs/1708.07747

Yann LeCun, Corinna Cortes, and C. J. B. (2010). *THE MNIST DATABASE of handwritten digits*. http://yann.lecun.com/exdb/mnist/