

Sistemas Inteligentes Híbridos

HYBRID SYSTEM COURSE PROJECT, MASTER SCIENCE DEGREE PROGRAM, CIN/UFPE/JUNE/2016

1

Improving Image Classification Accuracy using Hybrid Systems of Support Vector Machines and Convolutional Neural Networks

David Macêdo, *Member, IEEE*



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO

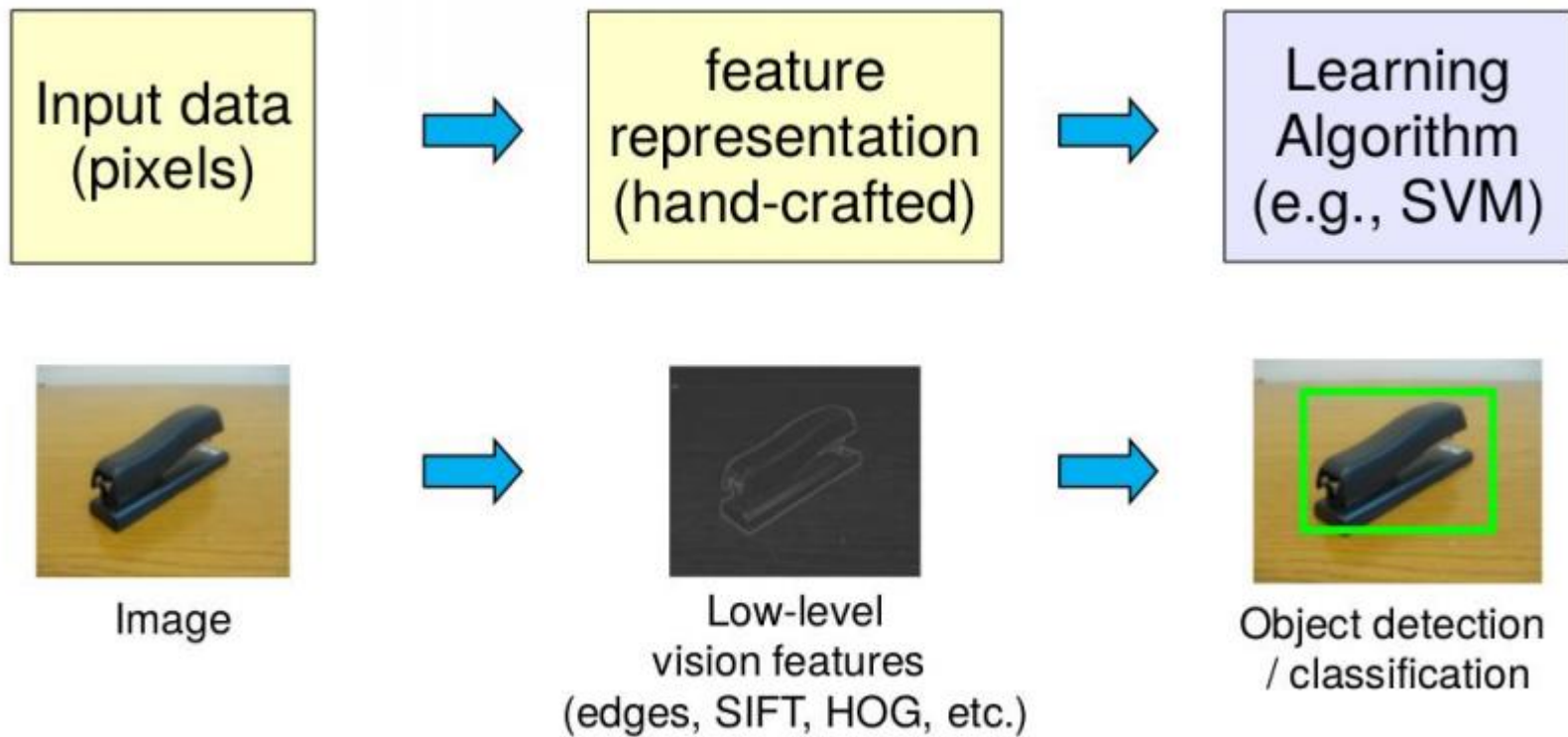


Motivation

- ➔ Present the limitations of SVMs (more generally, of shallow architectures) and CNNs
- ➔ Show that a Hybrid Systems combining SVMs and CNNs can achieve a better test accuracy than both original system components



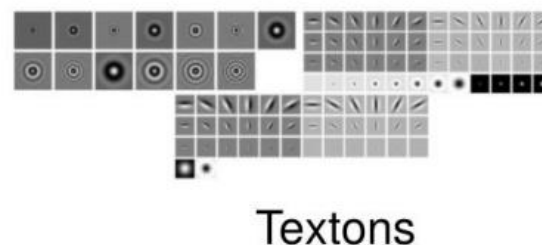
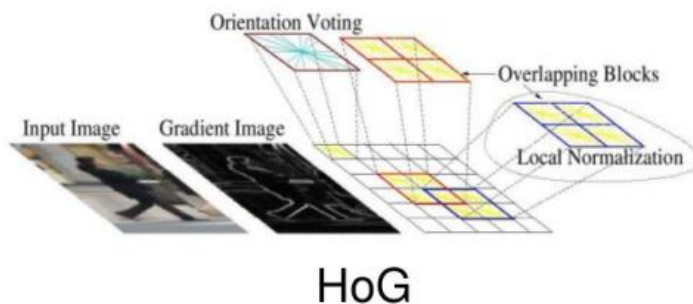
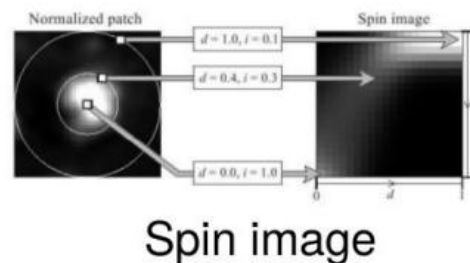
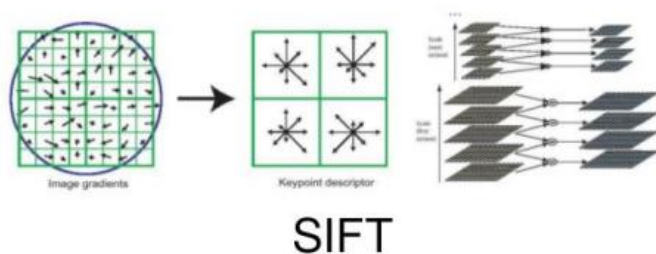
Support Vector Machines





Support Vector Machines

Computer vision features



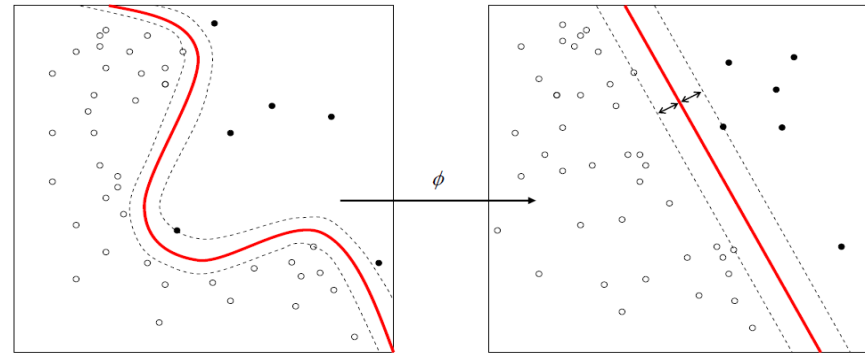
and many others:

SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH,



Support Vector Machines

- ➔ Maximum Margin Classifiers
- ➔ Empirical Risk Minimization
- ➔ Convex Optimization
- ➔ Shallow Architecture
- ➔ Feature Engineering

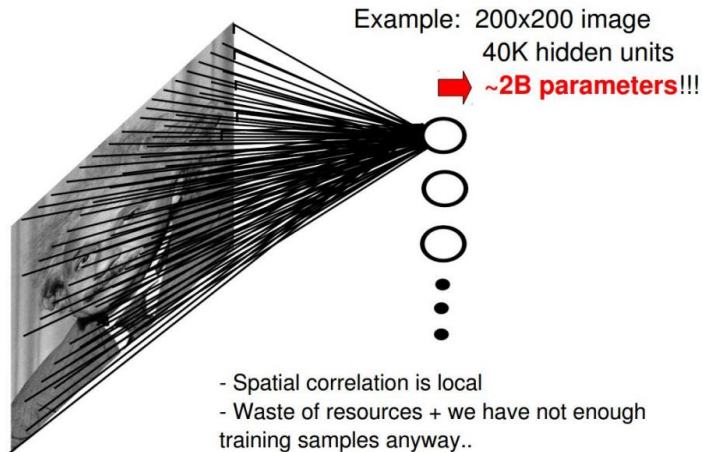


Good classifier, but do not learn deeper features from data

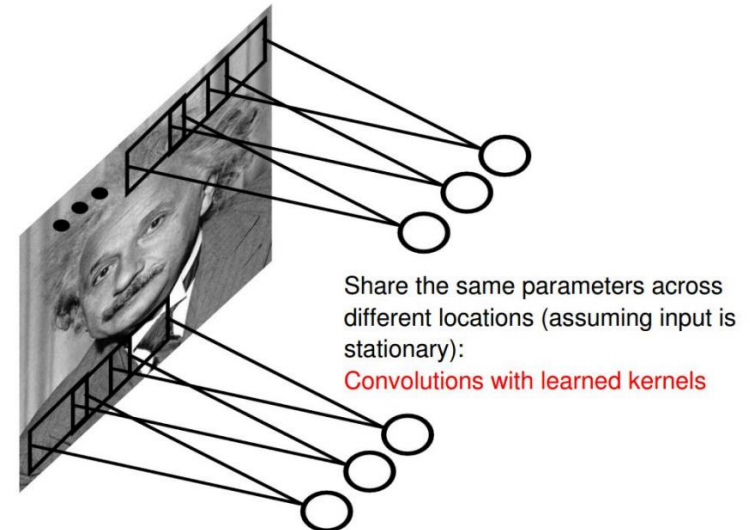


Convolutional Neural Networks

Fully Connected Layer



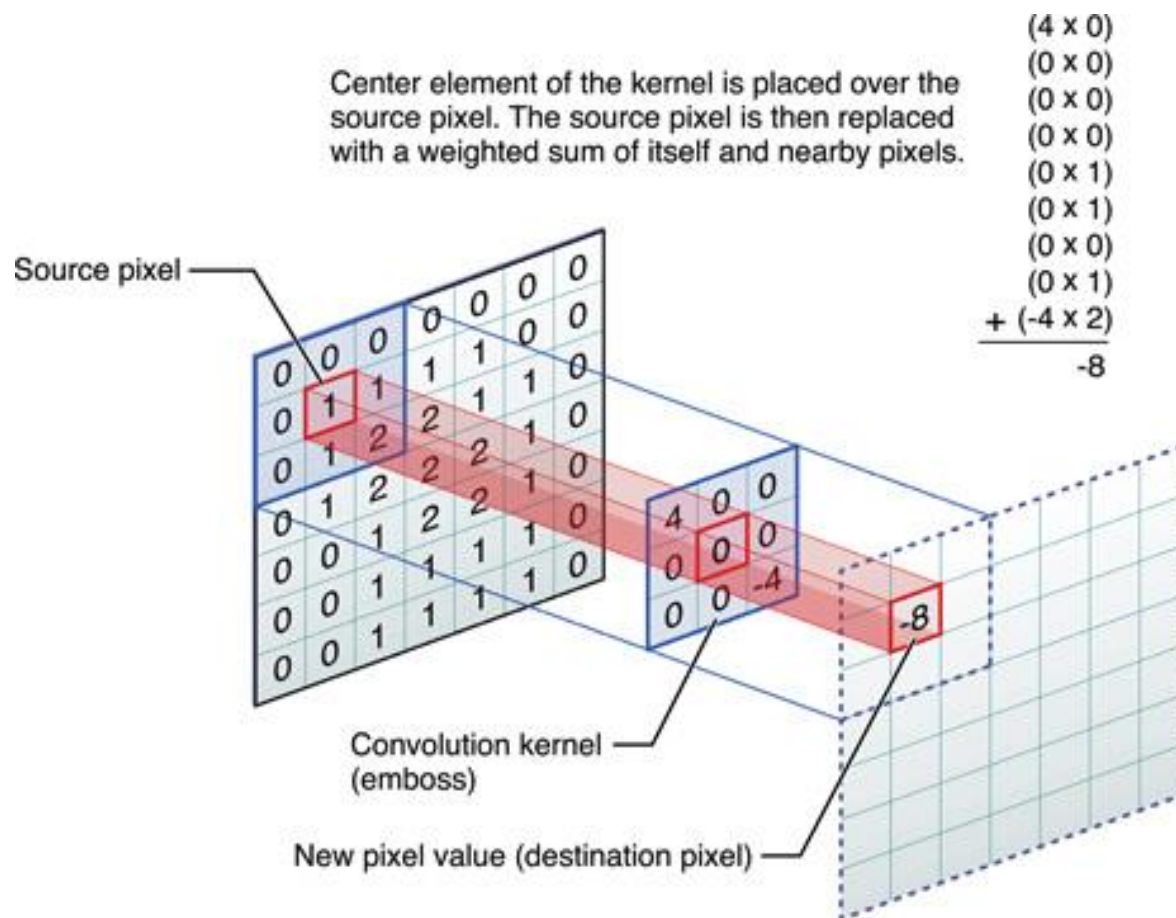
Convolutional Layer



Receptive Fields, Weight Sharing and Translational Symmetry: Feature Learning

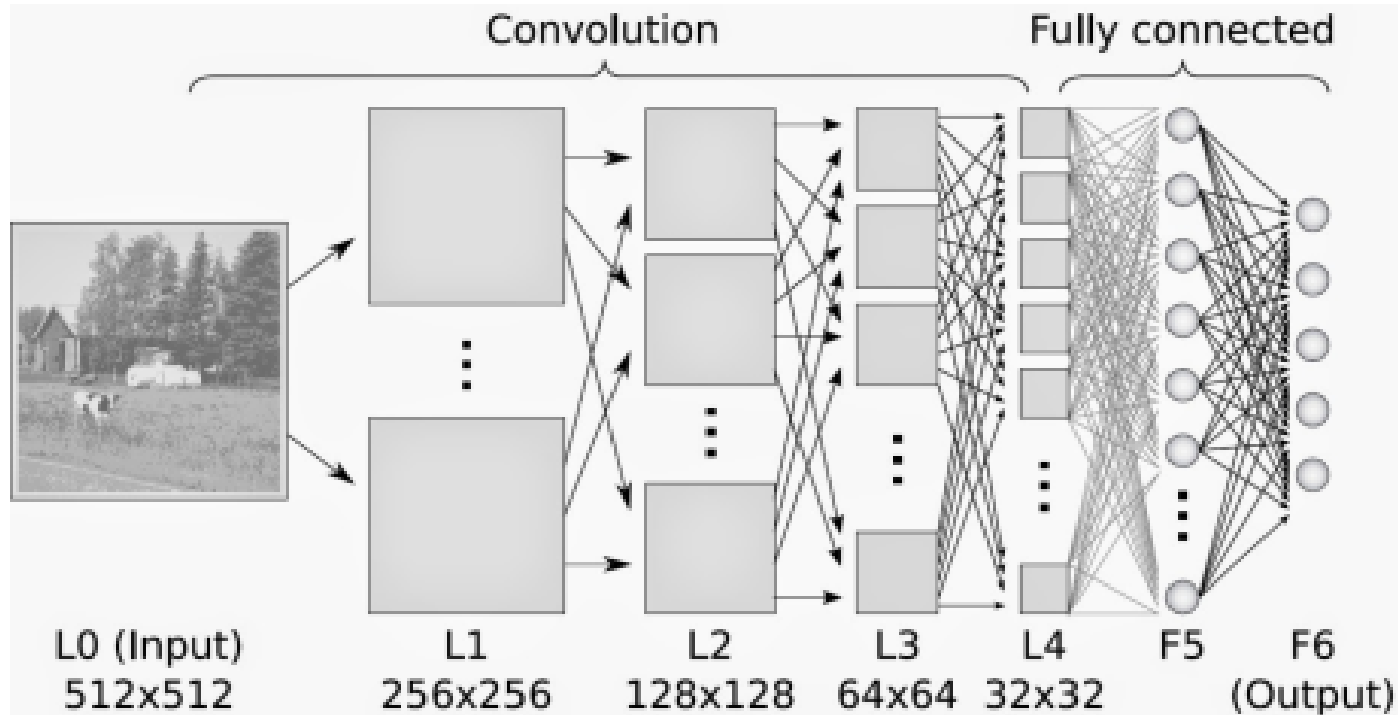


Convolutional Neural Networks





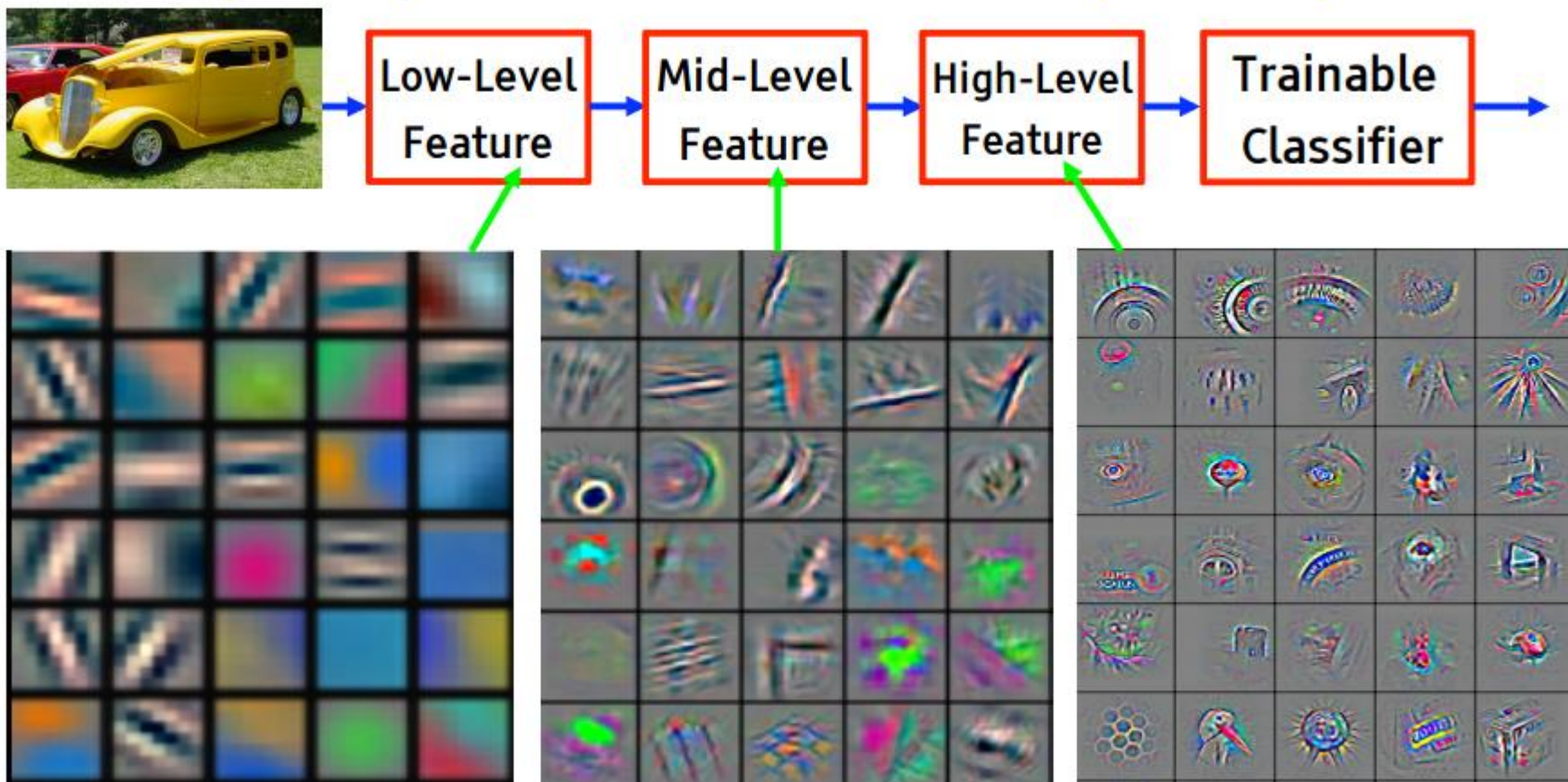
Convolutional Neural Networks



**Compositional and Hierarchical
Feature Learning Architecture**



Convolutional Neural Networks





Convolutional Neural Networks

- ➔ Hierarchical and Compositional Feature Learn
- ➔ Very Scalable and Parallel Computation
- ➔ “Perceptron-Like” Full Connected Classifier
- ➔ Non-convex Optimization

**Good learnable feature extractor,
but not so good classifier**



Proposed Hybrid System

- ➔ Train a Convolutional Neural Network (CNN)
- ➔ Remove the Last Full Connected Layer:
Extract the Features Learned
- ➔ Classifier the SVM using the Learned Features

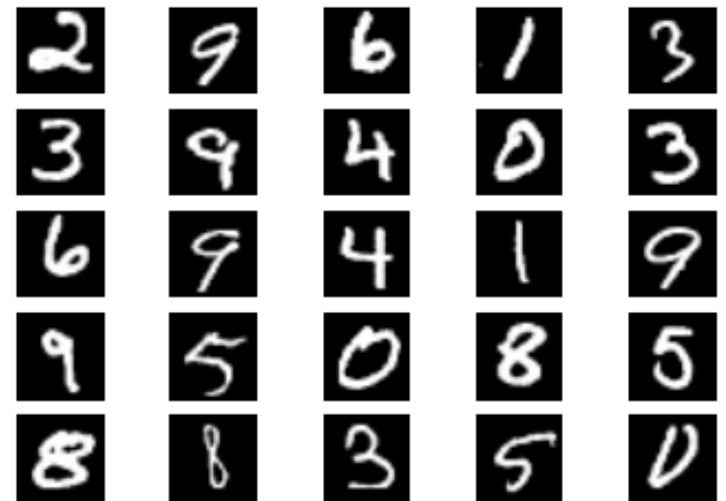
**The hope is to use the best characteristic
of each individual system in order to
create the hybrid system**



MNIST Dataset

- ➔ 28x28 Grayscale Images of Handwritten Digits
- ➔ Linearized 784 Features
- ➔ Training Set: 55.000 Examples
- ➔ Test Set: 10.000 Examples

Random Sampling of MNIST





Experiments: Models

- ➔ Linear Kernel SVM
- ➔ Gaussian Kernel SVM
- ➔ 1024 Nodes Extreme Learning Machine
- ➔ 4096 Nodes Extreme Learning Machine
- ➔ Convolutional Neural Network (CNN)
- ➔ SVM+CNN: The Hybrid Proposed System



Experiments: Models

- ➔ 5x5 Kernel Sizes in All Convolutional Layers
- ➔ The First Convolutional Layer = 32 Kernels
- ➔ The Second Convolutional Layer = 64 Filters
- ➔ The Third Layer = 128 Nodes Fully Connected
- ➔ The Fourth Layer = 10 Nodes Fully Connected



Experiments: Methods

- ➔ Each model were trained and tested when possible using a Graphics Processing Unit (GPU)
- ➔ **One hundred execution of training and test of each model was performed, with exception of SVM with Linear and Gaussian Kernels**
- ➔ **The extracted features are 128 higher level characteristics instead of the original 784 vector**



TABLE I
TEST PERCENT ACCURACY COMPARISON

Experiment	CNN	SVM _{CNN}
1	97.52	98.55
2	97.59	98.66
3	97.58	98.59
4	97.25	98.51
5	97.29	98.49
6	97.44	98.62
7	97.03	98.44
8	97.49	98.62
9	97.26	98.48
10	97.16	98.32

$$SVM_L = 93.93\%$$

$$SVM_G = 94.39\%$$

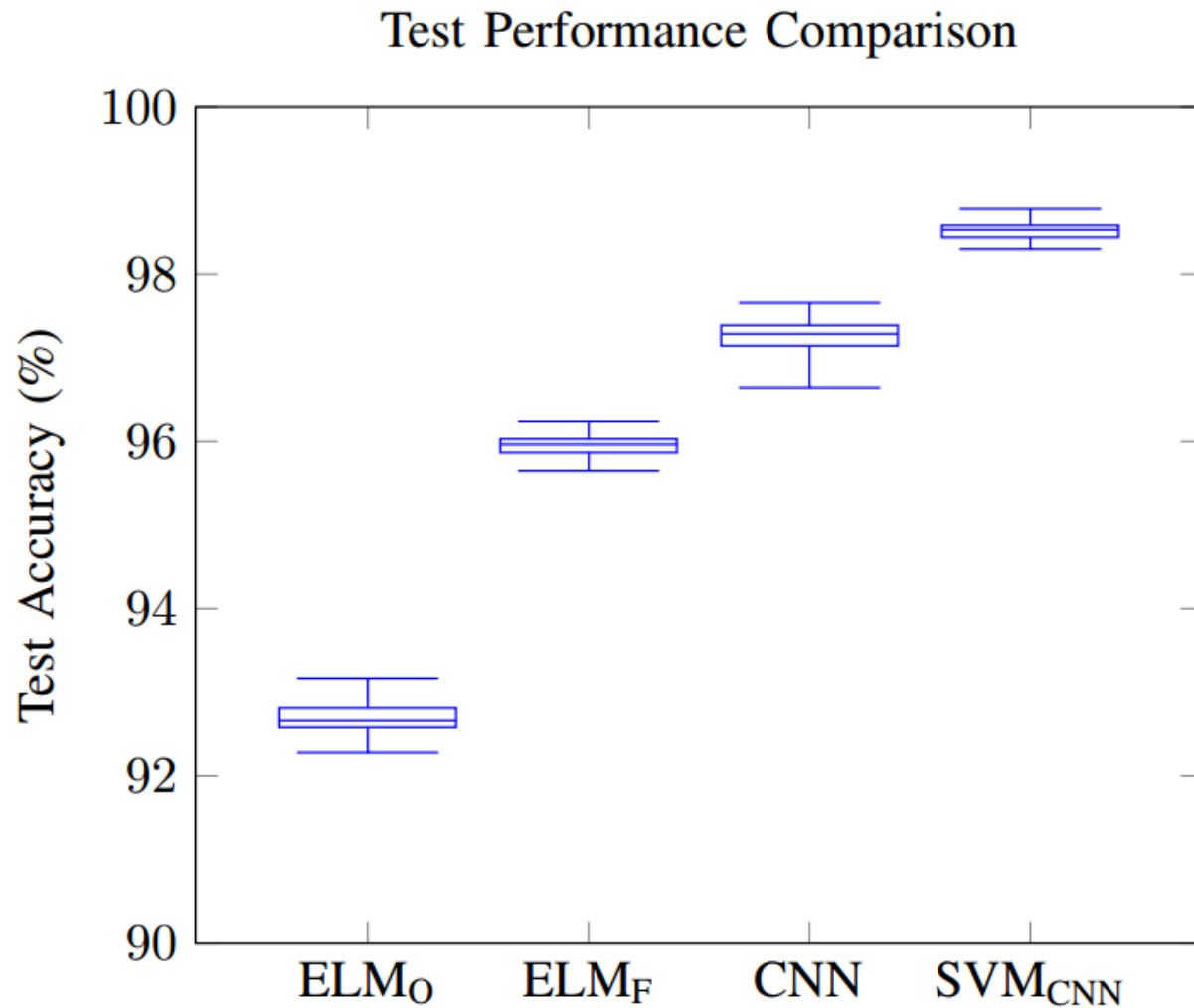


Fig. 1. Box plot of the distribution of test percent accuracy of one hundred of experiments of each model on MNIST data set.



Training Performance Comparison

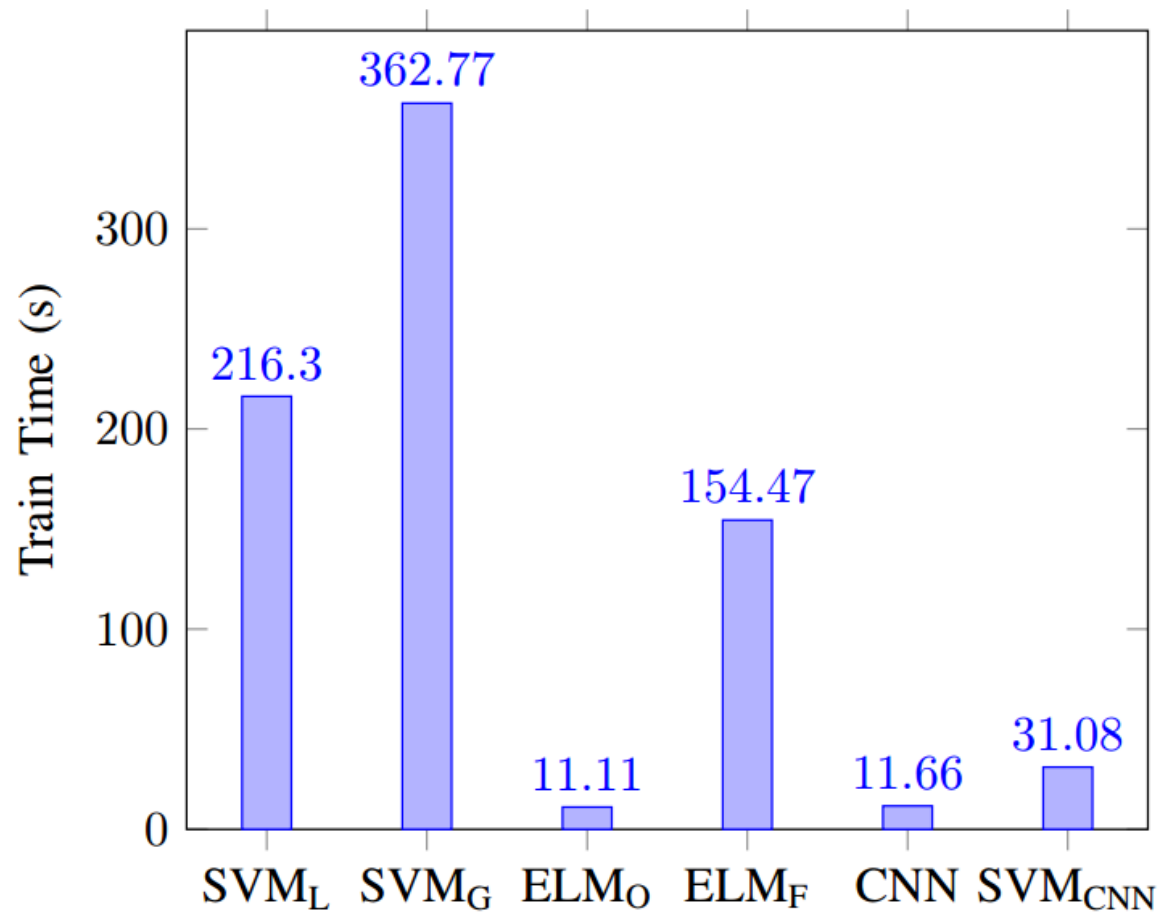


Fig. 2. Mean training time in seconds of one hundred of experiments of each model, with exception of SVM_L and SVM_G, that were executed just once since they have an almost deterministic training time.



Experiments: Models

- ➔ The proposed hybrid system achieves better test performance than both original systems
- ➔ The compound system presents much faster training time than the original SVM model
- ➔ The resultant system is very competitive when compared to state-of-art models



Software Tools



<https://github.com/dlmacedo/SVM-CNN>



References

- [1] V. N. Vapnik, Statistical learning theory. Wiley, 1998.
- [2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323, 1998.
- [5] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 9 1995. [Online]. Available: <http://link.springer.com/10.1007/BF00994018>
- [6] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 12 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and . Duchesnay, “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, no. Oct, pp. 2825–2830, 2011.