Classifying Objects at Different Sizes with Multi-Scale Stacked Sequential Learning

Eloi PUERTAS, Sergio ESCALERA and Oriol PUJOL [1, 2]

Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona
Centre de Visió per Computador

Abstract. Sequential learning is that discipline of machine learning that deals with dependent data. In this paper, we use the Multi-scale Stacked Sequential Learning approach (MSSL) to solve the task of pixel-wise classification based on contextual information. The main contribution of this work is a shifting technique applied during the testing phase that makes possible, thanks to template images, to classify objects at different sizes. The results show that the proposed method robustly classifies such objects capturing their spatial relationships.

Keywords. Sequential Learning, Multi-Scale Stacked Sequential Learning, Pixel-wise classification.

Introduction

Sequential learning [4] assumes that samples are not independently drawn from a joint distribution of the data samples $\mathbf{X}$ and their labels Y. In sequential learning the training data actually consists of sequences of pairs $(\mathbf{x}, y)$, so that neighboring examples display some correlation. Usually sequential learning applications consider one-dimensional relationship support, but this kind of relationships appear very frequently in other domains, such as images, or video. Consider the case of object recognition in image understanding. It is clear that if one pixel belongs to a certain object category, it is very likely that neighboring pixels also belong to the same object (with the exception of its borders).

In literature, sequential learning has been addressed from different perspectives: from the point of view of graphical models, using Hidden Markov Models or Conditional Random Fields (CRF) [8,7,5,12] for inferring the joint or conditional probability of the sequence. Graph Transformer Networks [2], considers the input and output as a graph and looks for the transformation that minimizes a loss function of the training data using a Neural Network. From the point of view of meta-learning sequential learning has been addressed by means of sliding window techniques, recurrent sliding windows [4] or stacked sequential learning (SSL) [3]. In our previous work [10], we identified that the

[1] Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain; E-mail: {*eloi,sergio,oriol*}*@maia.ub.es*

[2] Centre de Visió per Computador, Campus Universitat Autònoma de Barcelona, Edifici O, 08193, Bellaterra, Barcelona, Spain.

main step of the relationship modeling proposed in [3], is how the extended set is created. Thus we formalized a general framework for the SSL called MSSL where a multi-scale decomposition is used in such step.

In this work, we focus on pixel-wise classification based on contextual information. This is, to classify each pixel of an input testing image to a certain class. Generally, classes are objects inside the image or background. Such classes can be of any size in any context. General contextual classification aims to find and exploit any range of interactions. Observe that this concept depends explicitly on the notion of distance relative to the pattern of interest. The MSSL framework is able to learn such relationship between patterns implicitly as long as all the instances of such patterns holds similar relationships at the same range. However any successful sequential machine learning algorithm must be independent of this range value, at least while testing unseen instances. This is one of the big challenges in sequential learning. To address this problem, we propose the shifting technique applied at testing step. Thanks to this technique and the concept of template training set, the system becomes independent of the absolute range of interactions. Additionally, the MSSL framework is extended using a new multi-scale decomposition based on a multi-resolution approach that uses gaussian filters and a measure of likelihood for pixel classification is used instead of bare label predictions.

The paper is organized as follows: First we formalize the MSSL framework. Section 2 discusses the use of *shifting* technique for object classification at multiples scales. In the experimental section we test our system in two different scenarios and finally, last section concludes the paper and discusses future work.

1. Multiscale stacked sequential learning

SSL [3] is a meta-learning framework [11] consisting in two steps, first a base classifier is trained and tested with the original data. Then, an extended data set is created which joins the original training data features with the predicted labels produced by the base classifier considering a window around the example. Afterwards a second classifier is trained with this new feature set. In [10] SSL is generalized by emphasizing the key role of neighborhood relationship modeling. The framework presented includes a new block in the pipeline of the basic SSL. Figure 1 shows the Generalized Stacked Sequential

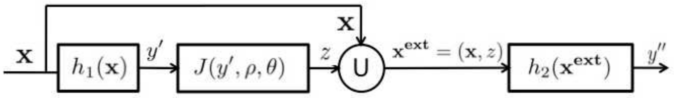
$\mathbf{x}$
$h_1(\mathbf{x})$
y'
$J(y', \rho, \theta)$
z
$\mathbf{x}$
U
$\mathbf{x}^{\mathbf{ext}} = (\mathbf{x}, z)$
$h_2(\mathbf{x}^{\mathbf{ext}})$
y''

Figure 1. Generalized stacked sequential learning.

Learning process. A classifier $h_1(x)$ is trained with the input data set $(\mathbf{x}, y)$ and the set of predicted labels y' is obtained. The next block defines the policy for creating the neighborhood model of the predicted labels. $z = J(y', \rho, \theta) : \mathcal{R} \to \mathcal{R}^w$ is a function that captures the data interaction with a model parameterized by θ in a neighborhood ρ. The result of this function is a w-dimensional value, where w is the number of elements in the support lattice of the neighborhood ρ. In the case of defining the neighborhood by means of a window, w is the number of elements in the window. Then, the output of $J(y', \rho, \theta)$ is joined with the original training data creating the extended training set $(\mathbf{x}^{\mathbf{ext}}, y) = ((\mathbf{x}, z), y)$. This new set is used to train a second classifier $h_2(\mathbf{x}^{\mathbf{ext}})$ with the goal of producing the final prediction y''. The proposed definition of $J(y', \rho, \theta)$ consists of two steps: first the multi-scale decomposition that answers how to model the relationship between neighboring locations, and second, the sampling that answers how to define the support lattice to produce the final set z.

The scale space is a very well-known tool for image analysis and processing. Its goal is to exploit the high correlation that exists in the neighboring pixels of an image and represent them in an efficient way. Observe that this goal is very similar to the objective of sequential learning in which we want to characterize and learn the relationship between examples according to their labels. We apply the idea of multi-scale decomposition upon predicted labels obtained by the first classifier. For the decomposition we use a multiresolution gaussian approach. Each level of the decomposition is generated by the convolution of the label field by a gaussian mask of variable σ, where σ defines the scale of the decomposition. This means that the bigger the sigma is, the longer interactions are considered. Thus, at each level of decomposition all the pixels have information from the rest, accordingly to the sigma parameter. Given a set of $\Sigma = \{\sigma_0, ..., \sigma_n\} \in \mathbb{R}^+$ and the predicted label sequence $\hat{y}^0(\mathbf{x})$ of length L, each level of the decomposition is computed as follows,

$$\hat{y}^{s_i}(\mathbf{x}) = g^{\sigma_i}(\mathbf{x}) * \hat{y}^0(\mathbf{x}) \tag{1}$$

where $g^{\sigma_i}(\mathbf{x})$ is defined as a multidimensional isotropic gaussian filter with zero mean,

$$g^{\sigma_i}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\sigma_i^{1/2}} e^{-\frac{1}{2}\mathbf{x}^T \sigma_i^{-1} \mathbf{x}} \tag{2}$$

Once we have the multi-scale decomposition, we define the support lattice. This is, the sampling performed over the multi-scale representation in order to obtain the extended data. Our choice is to use a scale-space sliding window over the multi-scale decomposition. The selected window has a fixed radius with length defined by r in each dimension and with origin in the current prediction example. Thus, the elements covered by the window is $w = (2r + 1)^d$ around the origin. For the sake of simplicity, we use a fixed radius of length $r = 1$. Then, for each scale i considered in the previous decomposition ($\sigma_i \quad i = 1 \dots n$), the window is stretched in each direction using a displacement proportional to the scale we are analyzing. In this paper we use a displacement $\delta_i = 3\sigma_i + 0.5$. This displacement at each scale forces that each point considered around the current prediction has very small influence from previous neighbor points. In this way, the number of features belonging to the extended data set is equal to $(2r+1)^d \times |\Sigma|$. Now, we can compute the value of $z_i = J(\hat{y}_i, \rho, \theta)$ defined above for the bi-dimensional case with $r = 1$ as follows,

$$z_i = \Big(\hat{y}^{s_0}_{(x-\delta_0,y-\delta_0)}, \hat{y}^{(s_0)}_{(x,y-\delta_0)}, \hat{y}^{(s_0)}_{(x+\delta_0,y-\delta_0)}, \dots, \hat{y}^{(s_0)}_{(x+\delta_0,y+\delta_0)}, \dots, \tag{3}$$

$$\hat{y}^{s_n}_{(x-\delta_n,y-\delta_n)}, \hat{y}^{(s_n)}_{(x,y-\delta_n)}, \hat{y}^{(s_n)}_{(x+\delta_n,y-\delta_n)}, \dots, \hat{y}^{(s_n)}_{(x+\delta_n,y+\delta_n)}\Big) \tag{4}$$

1.1. Extending the basic model: using likelihoods

In the MSSL model we use the predicted labels as the input of $J(y', \rho, \theta)$. An extension of this idea is to use a likelihood-based measure for each label instead of label prediction. The use of likelihoods gives a more precise information about the decisions of the first classifier than just its predictions. In the bi-class case, where the set of possible labels is $\mathcal{L} = \{\lambda_1, \lambda_2\}$, we have two membership likelihoods: $z = J(\{F(y = \lambda_1|x), F(y = \lambda_2|x)\}, \rho, \theta)$. The multi-scale decomposition and the sampling phases are the same, but now, each step is applied for each label, resulting in as many decomposition sequences as labels, and thus, the number of features in the extended set becomes $(2r+1)^d \times |\Sigma| \times |\mathcal{L}|$. This information can be taken into account by the second classifier, and then a more accurate prediction can be given, specially in those cases that the first classifier has few support for deciding the predicted label.

In order to obtain these values we need the base classifier $h_1(x)$ to generate not only a class prediction, but also its likelihood. Unfortunately, not all kind of classifiers can give a likelihood for its predictions. However, classifiers that work with margins such as Adaboost or SVM can be used [6]. In these cases, it is necessary to convert the margins used by these classifiers to a measure of likelihood. In case of using Adaboost, we apply a sigmoid function that normalizes Adaboost margins from the interval $[-\infty, \infty]$ to $[-1, 1]$ by means of the following equation, $f(x) = \frac{1-e^{-\beta m_x}}{1+e^{-\beta m_x}}$, where m_x is the margin given by Adaboost algorithm for the example x, and a constant that governs the transition: $\beta = \frac{-\ln(0.5\epsilon)}{0.25t}$. It depends on the number of iterations t that Adaboost performs, and an arbitrary small constant ϵ. Now, we use a soft distance to convert the normalized values to a likelihood in the range $[0, 1]$ for each label λ as follows: $f(x|y = \lambda_1) = e^{-\alpha d(-1, f(x))}$, $f(x|y = \lambda_2) = e^{-\alpha d(1, f(x))}$, where $\alpha = -\ln(\epsilon)/2$, and ϵ is an arbitrarily small constant (i.e $\epsilon = 10^{-3}$).

2. Learning at multiple scales

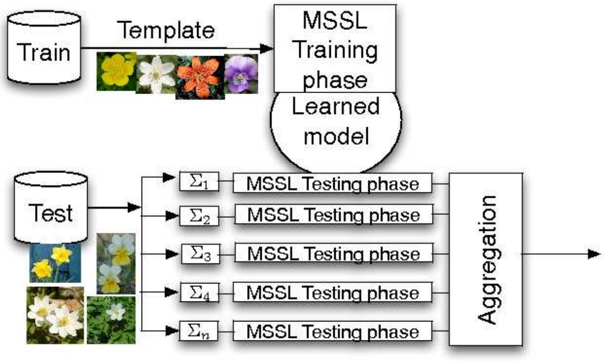

Figure 2. Architecture of the *shifting* technique.

In MSSL the choice of the scales is critical. The more scales are selected, the better performance is obtained. This is because different patterns at different scales can be de-

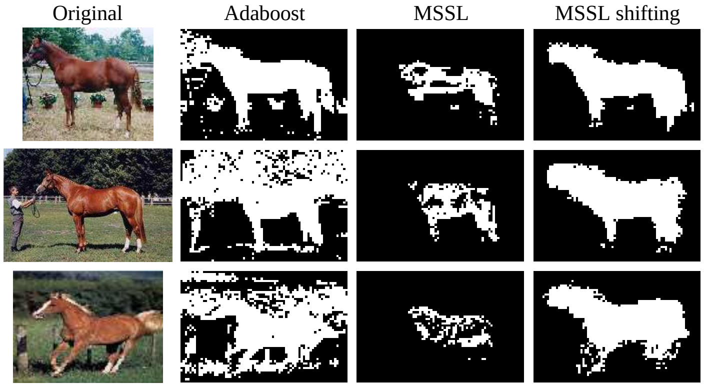

Figure 3. Examples of horse classification. Second column shows Adaboost prediction. Third and forth uses MSSL over the images with and without shifting.

tected. Nonetheless, if we learn a pattern with a concrete size, then when a new sequence at different size (smaller for example) is classified the prediction would not be correct using such scales. This is because the ranges of interactions displayed in the test sequence are not comparable with the ones displayed in the training phase, due to the fact that MSSL learns absolute interaction ranges. In order to effectively learn interactions in the pattern of interest we must ensure that the training set display these interactions at the same range. We call this particular training set a template. However, during the testing phase objects can be found at different sizes, thus displaying different interaction ranges than the ones that appear on the template. In order to successfully cope with this problem we propose an ensemble architecture at testing time. Figure 2 shows this architecture. It is based on the aggregation of the responses of the template trained system considering different relative range of interactions. If the interaction range set defined in MSSL by Σ follows a geometric progression $\sigma_i = k\sigma_{i-1}$, then testing at different ranges can be simply regarded as a shifting process of the extended features set. For example, given the features $\{y^{s_2}, y^{s_4}, y^{s_8}\}$ belonging to the extended set created during the training phase using template images of size (x, y) with $\Sigma_0 = \{2, 4, 8\}$ and a set of test images $\mathcal{X}$ of size $(x/2, y/2)$, then during the testing phase we use the same $\Sigma_0 = \{2, 4, 8\}$ and $\Sigma_1 = \{1, 2, 4\}$ we can observe that using the first Σ the features in both extended sets (training and testing) do not fit because the relationships between them are now halved. However, by using the second Σ now the test features have been shifted and they fit with those used for learning. Finally all results are combined with an aggregation function, for example taking the maximum value among all the likelihood responses for each sample.

3. Experiments and Results

In this section we test our methodology in two public databases consisting of horse and flower images [1,9].

3.1. Horse image classification using shifting

In order to validate our framework, first we define a toy problem using the Weizmann horse database [1], which consist in classify RGB horse images but rescaled to half size with respect to the ones used during the training phase. Each image is labelled according to the horse silhouette. We selected 100 images of horses from the database. Then, we define 5 random partitions of samples, each one consisting of the half of images for training and the remaining for testing. As a pre-processing step, we rescale all the horses images to the same resolution 150×100. The feature vector is composed of RGB attributes. All configurations use Adaboost with 100 iterations of decision stumps. For each image in the training set we perform a stratified sampling of 7500 pixels per image. This data is classified by the first base classifier applying leave-one-image-out. Using the generated predicted labels we perform a multi-scale decomposition with $\Sigma = \{2, 4, 8, 16\}$. The extended data set is created choosing the 8-neighbors of each pixel on each level of decomposition. Finally, both classifiers are trained using the same feature samples without and with the extended set, respectively. Table 1 shows results of predictions whether shifting is applied or not. For assessing the validity of the results we use the *Overlapping*, defined as $\frac{TP}{FN+FP+TP}$. First row shows the metrics using Adaboost. As sensitivity and specificity show up, the classification of the horses (*MSSL*) fails, because the system have learned the relative distance with respect to the size of the training horses. Now, we use the shifting approach by sliding the scales that are used in the testing phase to $\Sigma = \{1, 2, 4, 8\}$. As we can observe in Figure 3, applying this scale decomposition the model we trained before is able to classify the small horses appropriately without the need of retraining the system. Table 1 shows the improvement of the results classifying the small horses with the shifting approach.

Table 1. Results of prediction using Adaboost and MSSL with and without shifting technique.

Method	Acc	Over	Sens	Spec	Prec	NPV
Adatboost	0.7789	0.4417	0.8237	0.6559	0.8681	0.5749
MSSL	0.7465	0.0588	**0.9963**	0.0594	0.7444	**0.8561**
MSSL shift	**0.8734**	**0.6448**	0.8777	**0.8617**	**0.9458**	0.7193

3.2. Flowers classification using shifting

In this experiment we test our architecture in a free environment in which flowers can be found in different number and size [9]. For the training set, we define a flower template, this is, we select a group of similar flowers in size and shape, but from different types and colors. Each image is labelled according to the flower silhouette whether it is flower class or background class. For the testing set we choose flowers related to the defined template but at different size and color. We use 16 images for training and 25 for test. As a pre-processing step, we rescale all the images to the same resolution on the x-axis, maintaining the same proportion in the y-axis. The feature vector is only composed of RGB attributes. All configurations use Adaboost with 100 iterations of decision stumps.

For each image in the training set we perform a stratified sampling of 3000 pixels per image. This data is classified by the first base classifier applying leave-one-image-out. Using the generated predicted labels we perform a multi-scale decomposition with $\Sigma = \{18, 27, 41\}$. The extended data set is created choosing the 8-neighbors of each pixel on each level of decomposition. Finally, both classifiers are trained using the same feature samples without and with the extended set, respectively. We have performed several testing phases using always the same trained model. For each testing phase, we use a

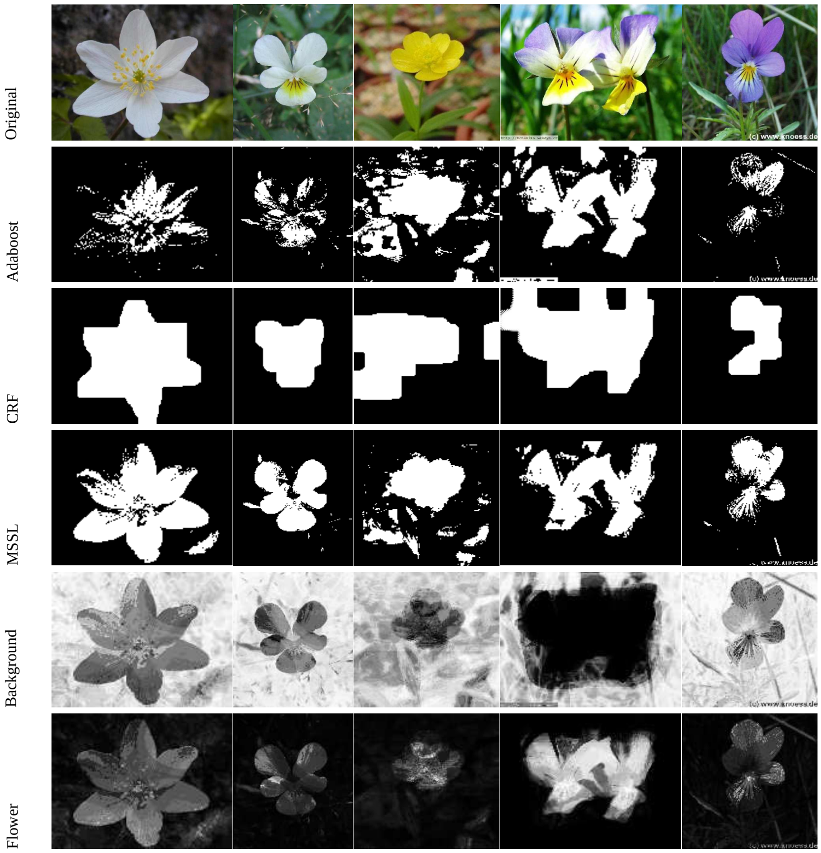

Figure 4. Predictions using Adaboost and MSSL.

Table 2. Results using Adaboost and MSSL.

Method	Acc	Over	Sens	Spec	Prec	NPV
ADABoost	0,8773	0,5621	0,9207	0,7217	0,9222	0,7176
CRF	0,8568	0,5840	0,8430	**0,9052**	**0,9689**	0,6220
MSSL	**0,9012**	**0,6243**	**0,9427**	0,7524	0,9317	**0,7858**

three scale decomposition from the range $\Sigma = \{0.5, 3, 5, 8, 12, 18, 27, 41\}$. This makes a total of 6 test rounds per image. At the end of each test round we take the measures of the likelihood of each image. Examples of background and flower likelihoods images at different rounds are shown in Figure 4. We calculate the maximum for all rounds, resulting in two images. The row *MSSL* shows the result of joining both images using the greater than operation. The figure also shows the original image and its resulting classification using Adaboost and CRF [8]. Table 2 shows the metrics for these methods. MSSL approach beats the non-sequential Adaboost approach for each metric and it also beats CRF in accuracy and overlapping. The rest of metrics point out that our method is better defining the flower class than the CRF method.

4. Conclusions

In this paper we adapted Multi-scale stacked sequential learning (MSSL) for classifying objects at different sizes. First, we introduced a gaussian mask for the creation of the multi-scale decomposition and a measure of likelihood for capturing spatial relations of data points. And second we proposed the *shifting* technique at testing time. This allows to correctly classify objects at different sizes than the learned ones. Results show the robustness and better performance of the presented methodology in comparison to classical approaches.

Acknowledgements

This work has been supported in part by projects TIN2009-14404-C02 and CONSOLIDER INGENIO CSD 2007-00018.

References

[1] E. Borenstein and S. Ullman, *Learning to segment*, ECCV (3) LNCS 3023, pp. 315–328, 2004.

[2] L. Bottou, Y. Bengio and Y.LeCun *Global training of document processing systems using graph transformer networks*, CVPR, pp. 489–494, 1997.

[3] W. W. Cohen and V. R. de Carvalho, *Stacked sequential learning*, IJCAI, pp. 671–676, 2005.

[4] T. Dietterich, ”Machine Learning for Sequential Data: A Review”, *SSSPR* vol. 2396, pp. 15-30, 2002.

[5] A. McCallum, D. Freitag, and F. Pereira, *Maximum entropy markov models for information extraction and segmentation*, Proc. of ICML 2000, pp. 591–598, 2000.

[6] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, *Additive logistic regression: a statistical view of boosting*, Annals of Statistics **28** (2000), 2000.

[7] J. D. Lafferty, A. McCallum, and F. Pereira, *CRF: Probabilistic models for segmenting and labeling sequence data*, Proc. of ICML 2001, pp. 282–289, 2001.

[8] C. H. Lee, R. Greiner, and M. Schmidt, *Support vector random fields for spatial classification*, PKDD, pp. 121–132, 2005.

[9] M. Nilsback and A. Zisserman, *Visual Vocabulary for Flower Classification*, CVPR, pp. 1447–1454, 2006.

[10] O. Pujol and E. Puertas and C. Gatta, *Multi-scale Stacked Sequential Learning*, MCS, pp. 262–271, 2009.

[11] D. H. Wolpert, *Stacked generalization*, Neural Networks, vol. 5, n. 2, pp. 241–259, 1992.

[12] T. G. Dietterich, A. Ashenfelter, and Y. Bulatov, *Training conditional random fields via gradient tree boosting*, In Proc. of the 21th ICML, 2004.