MARKOV RANDOM FIELDS MODELING IN ARTIFICIAL INTELLIGENCE

Cosmin Ţoca[1], Carmen Pătraşcu[2], Mihai Ciuc[3] and Dan-Alexandru Stoichescu[4]

The field of Artificial Intelligence has gained increased momentum in recent years, mainly due to the augmentation of the number of applications connected to the domain, such as object detection and classification. This article introduces two separate algorithms based on Markov Random Fields (MRF) for the previously mentioned applications. The solutions are validated by analyzing the results of the algorithm on state-of-the-art databases.

Keywords: pedestrian detection, object classification, Markov random fields

1. Introduction

Constant progress has been registered in recent years in the field of Artificial Intelligence, mainly due to the increased interest in two of its prominent applications, namely object detection and object recognition. Object detection represents an essential element in all engineering devices employing automated vision capabilities. Increased demands in the number of solutions and their success rates is motivated by the constant evolving number and complexity of related applications, such as advanced robotics, aided surveillance and automotive safety. The high complexity associated to finding objects in a wide variety of real world scenarios has lead to the necessity of using machine learning techniques, due to their ability of deriving and storing an implicit description of each object class from various pre-existing examples.

In the case of object classification, important progress was supported to a significant extent by the focus on the area of Deep Learning. The ability of integrating large neural networks into mass market appliances is already making a profound impact on the user experience.

[1] PhD., Faculty of Electronics, Telecommunications and Information Technology, University "Politehnica" of Bucharest, Romania, e-mail: cosmin.toca@gmail.com

[2] Associate Professor, Faculty of Electronics, Telecommunications and Information Technology, University "Politehnica" of Bucharest, Romania

[3] Professor, Faculty of Electronics, Telecommunications and Information Technology, University "Politehnica" of Bucharest, Romania

[4] Professor, Faculty of Electronics, Telecommunications and Information Technology, University "Politehnica" of Bucharest, Romania

This paper proposes the usage of Markov Random Fields [12, 15, 16] in the context of the previously mentioned applications.

With respect to object detection, we introduced a new set of channel features [5] which use the autobinomial Markov-Gibbs random fields [3]. Based on the presumption that information in images is spatially correlated, we solely employ close pixel neighborhoods for the probabilistic description of our object. To enhance the available information, we use both gradient magnitude and histograms from the image data.

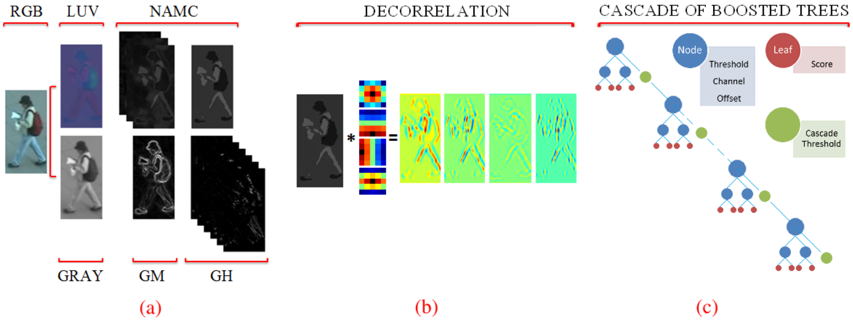


Fig. 1. Structure of the AutoMarkovian Object Detector [5, 4]. (a) Feature representation; (b) Local features' decorrelation; (c) Cascade of boosted trees

On the other hand, when considering neural networks, we can justify the usage of probabilistic graphical models by their common ability to predict effects derived from multiple causes. In the current structures of convolutional neural networks each neuron is connected only to a local region in the previous layer. This property builds a base for integrating Markov Random Fields (MRFs) into the structure of neurons in order to consider how likely a particular pathway is. We propose a novel type of convolutional layer based on Autobinomial Markov-Gibbs Random Fields, as shown in Fig. 2, which we call AutoMarkov Layer.

The remainder of this paper is organized as follows. Section 2 provides an introduction to the domain of Markov Random Fields and their properties. In Section 3 we describe the newly introduced channel features, giving a detailed overview of the autobinomial Markov channels. We describe the gradient magnitude and gradient histograms features, feature pyramids and the local decorrelation employed in our model. We also provide a detailed argumentation for the selection of the boosting method. In Section 4 we give a detailed overview of the proposed AutoMarkov Layers, describing the forward and backward propagation steps. Experimental evaluations are described separately for each specific application. Concluding remarks are presented in Section 5.

2. **Markov Random Fields**

Markov Random Fields (MRFs) are widely used probabilistic models providing a basis for modeling contextual constraints in visual processing and interpretation, as they allow the integration of prior knowledge extracted from images and scenes. This section focuses on generic Markov random fields models for natural images and their applications in object detection and classification.

We consider the image to represent a collection of random variables, indexed by their locations inside the image space, known as sites. Now, let us consider that $X = \{X_\xi\}_{\forall \xi \in \Omega}$ is a random process defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$ where Ω is a finite set that refers to the pixel's locations in the image configuration, with elements denoted by ξ and called sites. Let $\mathcal{F}$ be a finite set, called the phase space, that refers to all possible values of the pixels in a spectral plane and $\mathcal{P}$ the assignment of probabilities for each location.

A neighborhood system on Ω is a family $\mathcal{N} = \{\mathcal{N}_\xi\}_{\forall \xi \in \Omega}$ of subsets of Ω such that for all $\xi \in \Omega$ a site is not neighboring to itself $\xi \notin \mathcal{N}_\xi$, and the neighboring relationship is mutual $\eta \in \mathcal{N}_\xi \Leftrightarrow \xi \in \mathcal{N}_\eta$.

The Markov property states that each site ξ is conditionally independent of all other variables in the random field, except its neighboring system. The vicinity for each location can be described as:

$$\mathcal{N}_\xi = \left\{ \eta \in \Omega \mid \eta \neq \xi,\, \mathrm{d}^2(\xi, \eta) \leq \Delta \right\}, \tag{1}$$

where Δ is a fixed positive integer and $\mathrm{d}^2(\xi, \eta)$ is the squared Euclidean distance between ξ and η. To get the probability that at a site ξ the state is γ, we need to define a potential function $V_\mathcal{C}(\gamma)$ in the neighboring system, here denoted by a collection of cliques $\mathcal{C}$:

$$V_\mathcal{C}(\gamma) = \begin{cases} -\ln \begin{pmatrix} \Gamma \\ \gamma_\xi \end{pmatrix} + \gamma_\xi & \text{if } \mathcal{C} = \{\xi\} \\ \dfrac{\gamma_\xi \cdot \gamma_\eta}{\nu} & \text{if } \mathcal{C} = \{\xi, \eta\} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In this model, the only cliques participating in the energy function are singletons and pairs of mutual neighbors, the set of cliques appearing in the energy function being a disjoint sum of collections of cliques $\mathcal{C} = \sum_{k=1}^{\omega(\delta)} \mathcal{C}_k$ with $\omega(\delta)$ representing the number of possible cliques for each local specification.

The local property of a Markov Random field can be interpreted in terms of energy (Eq. 3) and potential (Eq. 2), leading to the following definition:

$$\pi^\xi = \frac{\mathrm{e}^{-\sum_{\mathcal{R}_\xi} V_\mathcal{R}(\gamma)}}{\sum\limits_{\varphi \in \mathcal{F}} \mathrm{e}^{-\sum_{\mathcal{R}_\xi} V_\mathcal{R}(\varphi, \gamma)}}, \tag{3}$$

where $V_\mathcal{R}$ is the potential function and $\varphi \in \mathcal{F}$.

3. **AutoMarkov Channels for Object Detection**

The algorithm employed for object detection applications proposed by the authors is based on a cascade of boosted trees, learned from a feature space consisting of several registered channel features. The motivation behind the usage of a boosted cascade of trees is to compute a series of weak classifiers in the form of simple regression trees, with each tree using the prediction residuals of the preceding levels. The authors propose learning a cascade of boosted trees from a feature space consisting of several registered channel features. By determining the best partitioning of the dataset at each step, the method separates the data into two sets at each split node, generating a set of binary trees. The next tree in the cascade will be fitted to the residuals representing the deviations of the previously observed values from the means. Based on the assumption that each partition represents the best scenario for the current stage, we ensure that the residual variance of the data is minimal. For the proposed algorithm, we keep a minimum depth of two levels for each binary tree, leading to a maximum of three decisional nodes and four leaves.

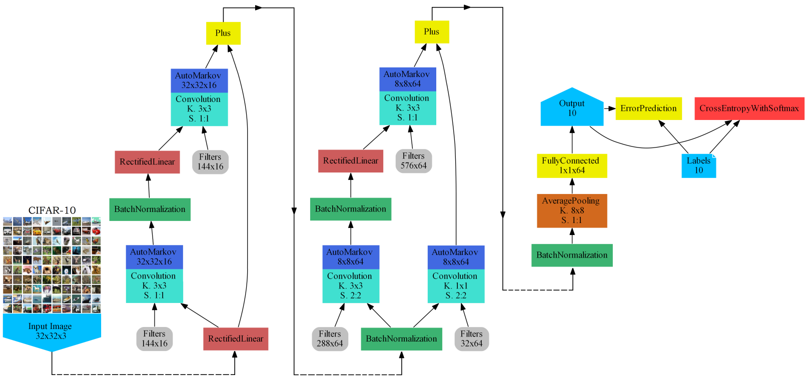


Fig. 2. Potential Architecture of Deep AutoMarkov Residual Network [6]. AutoMarkov Layers compute the response of a neuron as being the local property of a Markov Random Field.

A channel feature in this case represents a function of the same spatial variables as the initial image, meaning we can see it as an independent image. For feature calculation, if we replace Eq. 2 in Eq. 3, we get the probability assigned to the local system as:

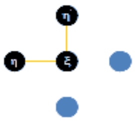
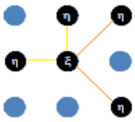
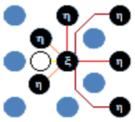
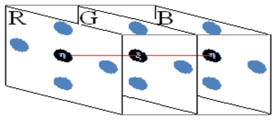
$$\pi^{\xi}(\gamma) = Z^{-1} \begin{pmatrix} \Gamma \\ \gamma_\xi \end{pmatrix} \sigma^{\gamma_\xi}(1-\sigma)^{\Gamma-\gamma_\xi}, \tag{4}$$

where $Z = \begin{pmatrix} \Gamma \\ \Gamma/2 \end{pmatrix}$ is a normalization constant, and $\sigma = \sigma(\mathcal{N}_\xi) = (\mathrm{e}^{\langle\alpha,\beta\rangle})/(1+\mathrm{e}^{\langle\alpha,\beta\rangle})$.

Here $\langle\alpha,\beta\rangle$ is the scalar product of two vectors α and β of sizes $\omega(\delta)$, and elements in α normalize the absolute difference of gray levels between pairs of cliques (as shown in Tab. 1) found in the same neighborhood of a certain pixel $\beta_k = \beta_k(\gamma_{\mathcal{N}_\xi}) = |\gamma_\eta - \gamma_{\eta'}|$, for $\eta \neq \eta'$, and $\eta, \eta' \in \mathcal{N}_\xi$, where $\{\eta, \xi\}$ and $\{\eta', \xi\}$ are two pairs in $\mathcal{C}_k$ containing ξ.

Table 1

Examples of neighbourhood systems. The maximum distance between two sites $\{\xi, \eta\}$ is denoted by δ and $\omega(\delta)$ and represents the number of possible cliques for each local specification, including only singletons and pairs of mutual neighbours.

	 (a)	3-by-3  (b)	5-by-5  (c)	 (d)
δ	1	2	4	1
$\omega(\delta)$	3	5	7	3

However, pixel values do not hold information only about the luminance intensity of color, but knowledge about features, such as weighted averages, gradient magnitudes and gradient orientations. In object detection applications it is considered good practice to extract multiple diverse features to get the maximum relevant information from the data. This is especially true for the case of person detection algorithms, where we focus on highlighting human body characteristics, regardless of the contextual information present in images.

It can easily be observed that the selection of channel features has a strong impact on the learning algorithm and is highly dependent on the application at hand. The highest influencing factors for the learning algorithm are the need for quick rejection or good performance measured at the end of the scanning process. In terms of features, we must take into consideration both the computational cost, as well as the memory space required to store all channel features at different scales. An important factor in the overall performance of the detector is the ability of each independent feature to separate between positive and negative classes.

The gradient magnitude ($\mathcal{M}(\gamma_\xi)$) at each site ξ is used to capture the undirected edge strength. To compute it we use a convolution with a discrete derivative mask in both horizontal and vertical directions, as can be seen in Eq. 5:

$$\mathcal{M}(\gamma_\xi) = \sqrt{\frac{\partial}{\partial x}\mathcal{N}_\xi^2 + \frac{\partial}{\partial y}\mathcal{N}_\xi^2}. \tag{5}$$

Using an inverse tangent function, we get the orientation of the gradients from the magnitude map, as seen in Eq. 6:

$$\mathcal{O}(\xi) = \arctan\left(\frac{\partial}{\partial y}\mathcal{N}_\xi / \frac{\partial}{\partial x}\mathcal{N}_\xi\right). \tag{6}$$

We can compute the channel features in the form of gradient histograms by weighting the gradient angles using the precomputed magnitudes. Based on the values from the gradient computation, each angle gets a weighted vote, as seen in Eq. 7:

$$\mathcal{H}_\theta(\xi) = \delta\Big(\mathcal{O}(\xi)\Big) \cdot \mathcal{M}(\xi), \tag{7}$$

where bin indexes can be determined by matching the orientation of each site to its corresponding angle:

$$\delta\Big(\mathcal{O}(\xi)\Big) = \begin{cases} 1, & \mathcal{O}(\xi) = \theta \\ 0, & \mathcal{O}(\xi) \neq \theta \end{cases} \tag{8}$$

The orientations are quantized in a fixed number of bins, leading to a set of six histogram channels computed from the luminance plane. The most important advantage of this approach is the usage of spatial distribution of orientations at each pixel location, helping predict the behavior of gradients in resampled images without the need of using computationally expensive analytical derivations.

Highly correlated data allow the user to check multiple features during each stage, which translates to oblique splits in the case of our boosted cascade of decisional trees. In order to reduce the computational cost required for processing, orthogonal trees trained on decorrelated data are used. A set of 4 decorrelation filters of 5 × 5 pixes were learned from training sets for the AutoMarkov Channels and applied in both training and scanning stages. Moreover, in order to speed up the scanning process we only compute the channels for the native resolution of the input data, approximating another twenty four scales based on the feature pyramids.

3.1. Performance Evaluation of the AutoMarkov Channels

In the case of computer vision methods, the algorithms' performance is constantly improved by learning from the available data. For this reason, continuous evaluation and testing represents an important part of the algorithm design process, helping isolate the practical use-cases where the algorithm fails, thus refocusing the research efforts on the most difficult scenarios. Object detection evaluation is mostly affected by the low number of samples in publicly available datasets. Adding to this problem is the reduced range of scales, occlusions and pose variations, which increase the difficulty of measuring the performance of the algorithm in real-life scenarios.

Given these considerations, we focus on pedestrian detection in order to perform the quality evalution of the proposed algorithm. We employ two state-of-the-art datasets to evaluate the proposed algorithm. The INRIA Pedestrian

Dataset is used as a benchmark to assess various parameter settings, thus leading to a thorough analysis of the proposed feature model. The selection of this dataset is motivated by the existence of high quality annotations of pedestrians in diverse settings and scenarios.

We evaluate the performance of different algorithms on INRIA Pedestrian Dataset and show the results and some additional information about the features types used by each algorithm in Tab. 2.

To achieve a complete understanding of the algorithm's performance we use different color spaces, combinations of features computed on separate color planes or combinations between channels, as well as vary the smoothing factor and number of neighborhood systems.

Each of the algorithms can use one or more types of features such as Color Channels, Gradient Magnitude (GM), Gradient Histograms (GH) and AutoMarkov Channels. A smoothing may be applied across channels, and channel features may be Locally Decorrelated (LD) or not. The Miss Rates (MRs) have been obtained by averaging only the results in the $[4 \cdot 10^{-3} - 10^{-1}]$ range of False Positive (FP) rates.

Table 2

Detection on INRIA Pedestrian Dataset

	AutoMarkov Channels			Color Channels	GM	GH	Smooth	LD	Log-Avg.
	Gray	Color	Multispectral	3	1	6		×4	MR
	1	3	1						$[4 \cdot 10^{-3} - 10^{-1}]$ FP
ACF [7]				✓	✓	✓	✓		22.38%
LDCF [13]				✓	✓	✓	✓	✓	17.45%
AutoMarkov		✓	✓	✓	✓	✓	✓	✓	15.57%
Sp.Pool [14]	-	-	-	-	-	-	-	-	14.49%
Franken [11]	-	-	-	-	-	-	-	-	14.24%

However, to comply with the increased demands of the field and cover a large range of real scenarios, we also perform an extensive evaluation [4] using the Caltech Pedestrian Dataset. The usage of this dataset allows the authors to perform an in-depth evaluation used to better characterize and understand the algorithm's behavior in a wide range of situations.

Table 3

Detection on Caltech Pedestrian Dataset

	AutoMarkov Channels			Color Channels	GM	GH	Smooth	LD	Log-Avg.
	Gray	Color	Multispectral	3	1	6		×4	MR
	1	3	1						$[4 \cdot 10^{-3} - 10^{-1}]$ FP
ACF [7]				✓	✓	✓	✓		86.00%
Franken [11]	-	-	-	-	-	-	-	-	85.27%
Roerei [1]	-	-	-	-	-	-	-	-	84.27%
AutoMarkov		✓	✓	✓	✓	✓	✓	✓	79.80%
Sp.Pool [14]	-	-	-	-	-	-	-	-	78.95%
LDCF [13]				✓	✓	✓	✓	✓	76.05%
Katamari [2]	-	-	-	-	-	-	-	-	78.86%

The results shown in Tab. 3 represents the most general test case with respect to the variability criterion, including all marked pedestrians in the Caltech Pedestrian Dataset.

We have shown that combining the AutoMarkov Channels together with gradient-based features in a cascade of boosted trees we outperform the majority of the existing features and methods for pedestrian detection.

Results computed with respect to the both datasets can be observed in the previous tables. When using the Caltech dataset, our results outperform the majority of the state-of-the-art solutions. For this scenario, our novel features are combined with gradient-based descriptors and a decorrelation algorithm is applied to each channel.

4. AutoMarkov Layers for Object Recognition

This article focuses on finding methods to improve the general outcome of a convolutional neural network, without limiting to specific problems. For this reason, we aim to add prior knowledge in specific points of a generic network configuration. Given the ability of the neurons in each convolutional layer to respect the Markov Property, we can employ probabilistic models. This means that the response of a neuron η is directly influenced only by the values of the receptive neurons $\mathcal{R}_\xi$, which implies that the neurons inside a layer Ω_η are only connected to a compact subset of the previous layer.

Considering the values of all the neurons in the receptive field, we can compute the response (π^ξ) of a neuron ξ as the local characteristic of a random field, defined as the probability of assigning to each neuron η a certain probability. Mapping the outputs in a probability space ensures that the learned filters β produce a stochastic response to a spatially local input.

The set of inputs of a certain layer is $\Omega_\xi = \mathbb{R}^{m \times n \times d}$ and the connections space is $\mathcal{F} : \mathbb{N} \to \mathbb{R}$, where $m, n, d \in \mathbb{N}$ and Γ_η represents the number of activations given by β_ξ in relation to the total number of inputs Γ_ξ. Generally speaking, a larger difference between the two numbers Γ_ξ and Γ_η accounts for an increase to the overall certainty level.

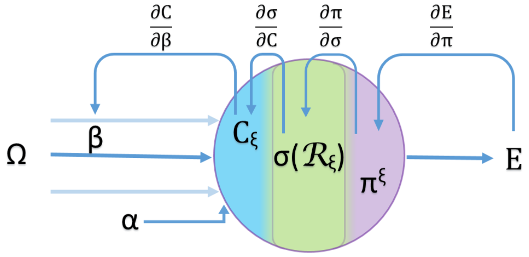

$\frac{\partial C}{\partial \beta}$
$\frac{\partial \sigma}{\partial C}$
$\frac{\partial \pi}{\partial \sigma}$
$\frac{\partial E}{\partial \pi}$
Ω
β
α
C_ξ
σ(ℛ_ξ)
π^ξ
E

Fig. 3. Forward and Backward Propagation. Simplified diagram of a neuron ξ from an AutoMarkov Layer. The response π^ξ is an autobinomial function with respect to the result of a convolution C_ξ between a filter β and the receptive field R_ξ, where E stands for the error of the output given a target response T_η and α is a bias.

We achieve the probability values needed in the Markovian probabilistic model by transforming the response of each convolutional layer. The results take into account both the receptive field and the certainty induced by systematic variations in the connections between adjacent layers.

After computing the values of the previous layer nodes, one can generate the activation value of the propagation's output as being the probability π^ξ assigned to a local system, as seen in Eq. 4. It should be mentioned that the forward step is shaped through a binomial distribution, modeling Γ_ξ independent success-failure experiments, each of them yielding success with probability $\sigma(\mathcal{R}_\xi)$. The sigmoid function $\sigma(\mathcal{R}_\xi)$ incorporates the convolution between the receptive field in Ω_ξ and the kernel field β, which allows to separately forward the convolution followed by the sigmoid and ending with the binomial mapping, as shown in Fig. 3.

We compute the derivative of the error function with respect to the weights $\partial E/\partial\beta_{\xi,\eta}$ to propagate the error $E = \frac{1}{2}(T_\eta - \pi^\xi)^2$ back through an AutoMarkov node. In this case, T_η represents the target output.

The derivative of the error function is easy to solve by using the chain rule that may be written using Leibniz's notation:

$$\frac{\partial E}{\partial\beta_{\xi,\eta}} = \frac{\partial E}{\partial\pi^\xi} \cdot \frac{\partial\pi^\xi}{\partial\sigma} \cdot \frac{\partial\sigma}{\partial\mathcal{C}_{\xi,\beta}} \cdot \frac{\partial\mathcal{C}_{\xi,\beta}}{\partial\beta_{\xi,\eta}} . \tag{9}$$

The derivative of the error with respect to the network output gives $\partial E/\partial\pi^\xi = -(T_\eta - \pi^\xi)$, while the derivative of the total network input with respect to the weights results in $\partial\mathcal{C}_{\xi,\beta}/\partial\beta_{\xi,\eta} = \mathcal{F}_\xi$, which are similar consequences to those of a regular convolutional layer.

The derivative of the sigmoid function σ with respect to $\mathcal{C}_{\xi,\beta}$ has been often used as typical activation function in many neural networks. As the sigmoid always has a positive derivative $\partial\sigma/\partial\mathcal{C}_{\xi,\beta} = \sigma(1-\sigma)$ the slope of the error function provides a descent direction which can be followed.

We perform a similar calculation to determine how the probability assigned to a local system π^ξ changes with respect to variations of $\sigma(\mathcal{R}_\xi)$:

$$\frac{\partial\pi^\xi}{\partial\sigma} = \pi^\xi \cdot \frac{\Gamma_\eta - \sigma\cdot\Gamma_\xi}{\sigma(1-\sigma)} . \tag{10}$$

Reducing the multiplication between the two middle terms by a common factor, we get:

$$\frac{\partial E}{\partial\beta_{\xi,\eta}} = \mathcal{F}_\xi \cdot (\sigma\Gamma_\xi - \Gamma_\eta) \cdot \pi^\xi (T_\eta - \pi^\xi) . \tag{11}$$

This section shows how to integrate AutoMarkov layers in a multi-layer convolutional neural network. It is important to note that the computation of the error terms must proceed backwards through the whole network, beginning with the output layer and ending with the first hidden layer.

4.1. Performance Evaluation of the AutoMarkov Layers

To have a complete overview of the proposed solution, the authors perform the functional testing of the algorithms on MNIST [10], CIFAR-10 [9] and CIFAR-100 [9] datasets. The aim of the testing is to assess and analyze the performance gain if the standard convolutions are replaced with the probabilistic model proposed in the present article by keeping the same network architecture (in terms of hyper-parameters) and the same configuration for the learning method (momentum, learning rate and batch size) as advised by the authors. For these experiments, the models have been trained for ten hours, with each training output evaluated on the corresponding testing dataset. We retain the classification error in Tab. 4, Tab. 5 and Tab. 6.

Table 4

Classification on MNIST. Comparison between standard Convolution Layers - AutoMarkov Layers within ResNet models on 28-by-28 gray level images divided into 10 classes.

	MNIST Testing Error - Top 1st	
	Regular Convolutions	AutoMarkov Layers
ResNet-56 [8]	0.59%	0.43%

Table 5

Classification on CIFAR-10. Comparison between standard Convolution Layers - proposed AutoMarkov Layers within ResNet models on 32-by-32 color images divided into 10 classes.

	CIFAR-10 Testing Error - Top 1st	
	Regular Convolutions	AutoMarkov Layers
ResNet-20 [8]	8.26%	7.83%
ResNet-56 [8]	6.43%	5.55%

Table 6

Classification on CIFAR-100. Comparison between standard Convolution Layers - proposed AutoMarkov Layers within ResNet models on 32-by-32 color images divided into 100 classes

	CIFAR-100 Testing Error - Top 1st	
	Regular Convolutions	AutoMarkov Layers
ResNet-56 [8]	28.29%	27.61%

Testing the proposed neuronal architecture within a configuration of ResNet with 56 layers on the MNIST database of handwritten digits offered an improvement of 0.16% and a correct classification rate of 99.57%.

In the case of the same model, the proposed convolutions led to an improvement from 93.57% to 94.45%, corresponding to 3rd position on CIFAR-10, while in the case of a shallow model composed of 20 convolutional layers, the measured improvement was 0.45%.

If the number of classes is much bigger, as is the case of CIFAR-100, the improvements brought by the proposed convolutional layers have increased up to 0.68%.

Deep neural networks usually have a large number of parameters detailing the complicated relationships between their inputs and outputs. With limited training data, these relationships will be the result of sampling noise leading to overfitting. If the network is just large enough to provide an adequate fit, it is unlikely for the model to overfit the training data, but the computation of a network's depth for specific scenarios represents an extremely difficult problem. Our results prove that the proposed AutoMarkov Layers improve the network generalization making it less prone to overfitting.

5. Conclusions

This work has been motivated by the inherent characteristic of natural images, namely the fact that adjacent pixel values are correlated. This characteristic allowed us to use probabilistic methods that, by their nature, are suitable for such problems. More precisely, we proposed the use of Markov Random Fields both for object detection and classification.

The object detection task deals with detecting instances of semantic objects of a certain class and is usually done by sliding window detectors in correlation with hand-crafted feature descriptors. The main advantage of using the proposed AutoMarkov Channels as feature descriptor comes from the property of randomly selecting pixels to be part of the neighborhood system, having a significant contribution for pedestrian detection in noisy scenarios. We have shown that combining the AutoMarkov Channels together with gradient-based features in a cascade of boosted trees and using a method that substitutes the need for oblique splits, we outperform the majority of the existing features and methods for pedestrian detection.

The object classification task is accomplished by a novel type of convolutional layer based on Autobinomial Markov-Gibbs Random Fields, called the AutoMarkov Layer. We have exploit the standardized architectures of various configurations of deep neural networks developed in recent years and added prior knowledge to the neurons by integrating a probabilistic model which focuses on their interaction and computes probabilities associated to each particular pathway.

The functional testing is performed using the MNIST, CIFAR-10 and CIFAR-100 datasets. By replacing the Standard Convolutional Layers with the AutoMarkov Layers have been measure improvements of up to 1% correct classification on the previously mentioned datasets leading to a correct classification rate of 99.57% on MNIST, 94.45% on CIFAR-10 corresponding to 3rd position on the object classification benchmark and respective 72.39% on CIFAR-100.

Acknowledgement

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/134398.

REFERENCES

[1] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool. Seeking the strongest rigid detector. In *CVPR*, 2013.

[2] Rodrigo Benenson, Mohamed Omran, Jan Hendrik Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned? *CoRR*, 1411.4304, 2014.

[3] Pierre Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues.* Texts in Applied Mathematics. Springer, 1999.

[4] C. Ţoca, C. Pătraşcu, and M. Ciuc. Performance testing and functional limitations of normalized autobinomial markov channels. In *Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on*, pages 401–405, September 2015.

[5] Cosmin Ţoca, Mihai Ciuc, and Carmen Pătraşcu. Normalized autobinomial markov channels for pedestrian detection. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 175.1–175.13. BMVA Press, September 2015.

[6] Cosmin Ţoca, Mihai Ciuc, and Carmen Pătraşcu. Automarkov dnns for image classification. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, December 2016.

[7] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1532–1545, Aug 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] Markus Mathias, Rodrigo Benenson, Radu Timofte, and Luc Van Gool. Handling occlusions with franken-classifiers. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1505–1512. IEEE, 2013.

[12] J. Moussouris. Gibbs and markov random systems with constraints. *Journal of Statistical Physics*, 10(1):11–33, 1974.

[13] Woonhyun Nam, Piotr Dollar, and Joon Hee Han. Local decorrelation for improved pedestrian detection. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 424–432. Curran Associates, Inc., 2014.

[14] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton van den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision ECCV 2014*, volume 8692 of *Lecture Notes in Computer Science*, pages 546–561. Springer International Publishing, 2014.

[15] F. Preston. Gibbs states on countable sets. *Cambridge University Press*, 1974.

[16] W. Woods. Two-dimensional discrete markovian fields. *IEEE Transactions on Information Theory*, 18(2):721741, 1972.