Goal!! Event detection in sports video

Grigorios Tsagkatakis [†]***, Mustafa Jaber*** [⋆]***, Panagiotis Tsakalides*** [†‡]
[†] ***Institute of Computer Science, Foundation for Research & Technology - Hellas (FORTH), Crete, 73100, Greece.***
[⋆] ***NantVision Inc., Culver City, CA, 90230, USA.***
[‡] ***Department of Computer Science, University of Crete, Crete, 73100, Greece.***

Abstract

Understanding complex events from unstructured video, like scoring a goal in a football game, is an extremely challenging task due to the dynamics, complexity and variation of video sequences. In this work, we attack this problem exploiting the capabilities of the recently developed framework of deep learning. We consider independently encoding spatial and temporal information via convolutional neural networks and fusion of features via regularized Autoencoders. To demonstrate the capacities of the proposed scheme, a new dataset is compiled, compose of goal and no-goal sequences. Experimental results demonstrate that extremely high classification accuracy can be achieved, from a dramatically limited number of examples, by leveraging pre-trained models with fine-tuned fusion of spatio-temporal features.

Introduction

Analyzing unstructured video streams is a challenging task for multiple reasons [10]. A first challenge is associated with the complexity of real world dynamics that are manifested in such video streams, including changes in viewpoint, illumination and quality. In addition, while annotated image dataset are prevalent, a smaller number of labeled datasets are available for video analytics. Last, the analysis of massive, high dimensional video streams is extremely demanding, requiring significantly higher computational resources compared to still imagery [11].

In this work, we focus on the analysis of a particular type of videos showing multi-person sport actives and more specifically football (soccer) games. Sport videos in general are acquired from different vantage points and the decision of selecting a single stream for broadcasting is taken by the director. As a result, the broadcasted video stream is characterized by varying acquisition conditions like zooming-in near the goalpost during a goal and zooming-out to cover the full field. In this complex situation, we consider the high level objective of detecting specific and semantically meaningful events like an opponent team scoring a goal. Succeeding in this task will allow the automatic transcription of games, video summarization and automatic statistical analysis.

Despite the many challenges associated with video analytics, the human brain is able to extract meaning and provide contextual information in a limited amount of time and from a limited set of training examples. From a computational perspective, the process of event detection in a video sequence amounts to two foundamental steps, (i) spatio-temporal feature extraction and (ii) example classification. Typically, feature extraction approaches rely on highly engineered handcrafted features like the SIFT, which however are not able to generalize to more challenging cases. To achieve this objective, we consider the state-of-the-art framework of deep learning [18] and more speciically the case of Convolutional Neural Networks (CNNs) [16], which has taken by storm almost all problems related to computer vision, ranging from image classification [15, 16], to object detection [17], and multimodal learning [6]. At the same time, the concept of Autoencoders, a type of neural network which tries to appropriate the input at the output via regularization with various constrains, is also attracting attention due to its learning capacity in cases of unsupervised learning [21].

While significant effort has been applied in design and evaluating deep learning architectures for image analysis, leading to highly optimized architectures, the problem of video analysis is at the forefront of research, where multiple avenues are explored. The urgent need for video analytics is driven by both the wealth of unstructured videos available online, as well as the complexities associated with adding the temporal dimension. In this work we consider the problem of goal detection in broadcasted low quality football videos. The problem is formulated as a binary classification of short video sequences which are encoded though a spatio-temporal deep feature learning network. The key novelties of this work are:

- Develop a novel dataset for event detection in sports video and more specifically, for goal detection is football games.
- Investigate deep learning architectures, such as CNN and Autoencoders, for achieving efficient event detection.
- Demonstrate that learning, and thus accurate event detection can be achieved by leveraging information from a few labeled examples, exploiting pre-trained models.

State-of-the-art

For video analytics, two major lines of research have been proposed, namely frame-based and motion-based, where in the former case, features are extracted from individual frames, while in the latter case, additional information regarding the inter-frame motion like optical flow [3] is also introduced.

In terms of single frame spatial feature extraction, CNNs have had a profound impact in image recognition, scene classification, and object detection among others [16]. To account to the dynamic nature of video, a recently proposed concept involves extenting the two-dimensional convolution to three dimensions, leading to $3D$ CNNs, where temporal information is included as a distinct input [12, 13]. An alternative approach for encoding the temporal informarion is through the use of Long-Short Term Memory (LSTM) networks [1, 13], while another concept involves the generation of *dynamic images* through the collapse of multiple video frame and the use of $2D$ deep feature exaction on such representations [7]. In [2], temporal information is encoded through average pooling of frame-based descriptors and

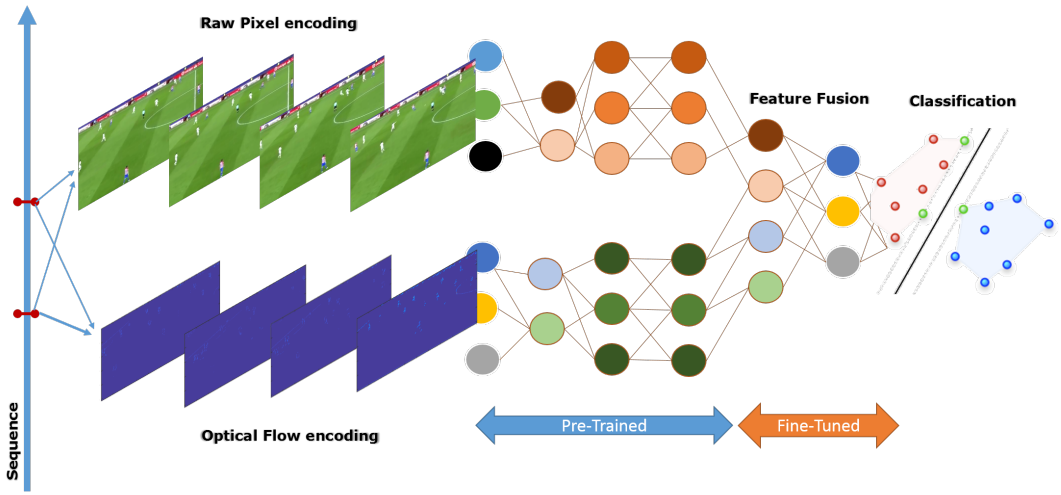

Figure 1: Block diagram of the proposed Goal detection framework. A 20-frame moving window initially selects part of the sequence of interest, and the selected frames undergo motion estimation. Raw pixel values and optical flows are first independently encoded using the pre-trained deep CNN for extracting spatial and temporal features. The extracted features can either be introduced into a higher level network for fusion which is fine-tuned for the classification problem, or concatenated and used as extended input features for the classification.

the subsequent encoding in Fisher and VLAD vectors. In [4], the authors investigated deep video representation for action recognition, where temporal information was introduced in the *frame-diff* layer of the deep network architecture, through different temporal pooling strategies applied in patch-level, frame-level and temporal window-level.

One of the most sucessfull framework for encoding both spatial and temploral information is the two-stream CNN [8]. Two-streams networks consider two source of information, raw frames and optical flow, which are independently encoded by a CNN and fused into an SVM classifier. Further studies on this framework demonstrated that using pre-trained models can have a dramatic impact on training time, for the spatial and temporal features [22], while convolutional two-stream network fusion was recently applied in video action recognition [23]. The combination of $3D$ convolutions and the two-stream approach was also recently reported for video classification, achieving state-of-the-art performance at significantly lower processing times [24]. The performance demonstrated by the two-streams approach for video analysis led to the choice of this paradigm in this work.

Event Detection Network

The proposed temporal event detection network is modeled as a two-stream deep network, coupled with a sparsity regularized Autoencoder for fusion of spatial and temporal data. We investigate Convolutional and Autoencoder Neural Networks for the extraction of spatial, temporal and fused spatio-temporal features and the subsequent application of kernel based Support Vector Machines for the binary detection of goal events. A high level overview of the processing pipeline is shown in Figure 1.

While in fully-connected networks each hidden activation is computed by multiplying the entire input by the corresponding weights in that layer, in CNNs each hidden activation is computed by multiplying a small local input against the weights. The typical structure of a CNN consists of a number of convolution and pooling/subsampling layers, optionally followed by fully connected layers. At each convolution layer, the outputs of the previous layer are convolved with learnable kernels and passed through the activation function to form this layer's output feature map.

Let $n \times n$ be a square region extracted from a training input image $\mathbf{X} \in \mathbb{R}^{N \times M}$, and $\mathbf{w}$ be a filter of kernel size $(m \times m)$. The output of the convolutional layer $\mathbf{h} \in \mathbb{R}^{(n-m+1)\times(n-m+1)}$ is given by:

$$\mathbf{h}_{ij}^{\ell} = \sigma\Big(\sum_{a=0}^{m-1}\sum_{b=0}^{m-1} \mathbf{w}_{ab}\mathbf{x}_{(i+a)(j+b)}^{\ell-1} + \mathbf{b}_{ij}^{\ell}\Big), \tag{1}$$

where $\mathbf{b}$ is the additive bias term, and $\sigma(\cdot)$ stands for the neuron's activation unit. Specifically, the activation function σ, is a standard way to model a neurons output, as a function of its input. Convenient choices for the activation function include the logistic sigmoid, the hyperbolic tangent, and the Rectified Linear Unit. Taking into consideration the training time required for the gradient descent process, the saturating (*i.e* tanh, and logistic sigmoid) non-linearities are much slower than the non-saturating ReLU function.

The output of the convolutional layer is directly utilized as input to a sub-sampling layer that produces downsampled versions of the input maps. There are several types of pooling, two common types of which are max-pooling and average-pooling which partition the input image into a set of non-overlapping or overlapping patches and output the maximum or average value for each such sub-region. For the $2D$ feature extraction networks, we consider the VGG-16 CNN architecture, which is composed of 13 convolutional layers, while five of them are followed by a max-pooling layers, leading to the three fully connected layers [9].

Unlike image detection problems, feature extraction in video must address the challenges associated with the variation of the duration of events, in addition to the challenges related to illumi-

nation and viewpoint variability. We fuse the two representations using a sparsity regularized Autoencoder and more specifically, we consider all available training data from all classes as input to the unsupervised Autoencoder to extracting features encoding both spatial and temporal information.

Formally, the Autoencoder is a deterministic feed-forward artificial neural network comprised of an input and an output layer of the same size with a hidden layer in between, which is trained with backpropagation in a fully *unsupervised* manner, aiming to learn an approximation $\hat{\mathbf{z}}$ of the input which would be ideally more descriptive of the the raw input. The feature mapping that transforms an input pattern $\mathbf{z} \in \mathbb{R}^n$ into a hidden representation $\mathbf{h}'$ (called code) of k neurons (units), is defined by the *encoder* function:

$$f(\mathbf{z}) = \mathbf{h}' = \alpha_f(W_1\mathbf{z} + \mathbf{b_1}), \tag{2}$$

where $\alpha_f : \mathbb{R} \mapsto \mathbb{R}$ is the *activation function* applied componentwise to the input vector. The activation function is usually chosen to be nonlinear; examples include the logistic sigmoid and the hyperbolic tangent.

The activation function is parametrized by a weight matrix $W_1 \in \mathbb{R}^{k\times n}$ with models the connections between the input and the hidden layer and a bias vector $\mathbf{b_1} \in \mathbb{R}^{k\times 1}$. The network output is then computed by mapping the resulting hidden representation $\mathbf{h}'$ back into a reconstructed vector $\hat{\mathbf{z}} \in \mathbb{R}^{n\times 1}$ using a separate *decoder* function of the form:

$$g(f(\mathbf{z})) = \hat{\mathbf{z}} = \alpha_g(W_2\mathbf{h}' + \mathbf{b_2}), \tag{3}$$

where α_g is the activation function, $W_2 \in \mathbb{R}^{n\times k}$ is the decoding matrix and $\mathbf{b_2} \in \mathbb{R}^n$ a vector of bias parameters which are learned from the hidden to the output layer.

The estimation of the parameters set $\theta = \{W_1, \mathbf{b_1}, W_2, \mathbf{b_2}\}$ of an Autoencoder, is achieved through the minimization of the reconstruction error between the input and the output according to a specific loss function. Given the training set $\mathbf{Z}$, a typical loss function seeks to minimize the normalized sum of squares error, defining the following optimization objective:

$$J_{\mathrm{AE}}(\theta) = \frac{1}{m}\sum_{i=1}^{m} \| \frac{1}{2} z^{(i)} - \hat{z}^{(i)} \|^2 + \mathscr{R}(z) \tag{4}$$

where $\hat{x}$ is implicitly dependent on the parameter set θ, $\|\cdot\|$ is the Euclidean distance and $\mathscr{R}(z)$ is a regularization term. Sparse Autoencoders are a special case of the traditional Autoencoders, where the code is constrained to be sparse, *i.e.* only a small fraction of hidden units are activated by the inputs. Signal and model sparsity have had a profound impact on signal processing and machine learning due to their numerous advantages, such as robustness, model complexity, generative and discriminative capabilities among others, e.g., [20].

Dataset

To evaluate the performance of the proposed sports event detection framework, a novel dataset was compiled using video sequences downloaded from a public repository (YouTube.com). For the "Goal" class, 200 sequences of 2-3 seconds videos where extracted, depicting the event of interest under a large number of viewing conditions including camera location, ego-motion, drastic illumination changes, low quality encoding, motion artifacts and cluttering among others. Similarly, for the "NoGoal" class, 200 sequences were also extracted from different games, showing different examples of confusing activities like near-miss shots and fouls. Representative examples of sequences from the dataset are shown in Figure 2.

Figure 2: Example of frames extracted from the Goals (left column) and No Goals (right column) sequences. One can easily observe that typically, sequences where a goal is scored are captured with a greater zoom factor compared to generic scenes, introducing artifacts like motion blurring.

Experimental Evaluation

For generating the reported experimental results, we consider spatial and temporal features from 20 color frames per sequences, extracted using the VGG-16 network. To introduce the temporal aspects, we employ the Lukas-Kanade optical flow estimation with threshold for noise reduction set to 0.009. To make the optical flow compatible with the pixel values, a linear normalization to the [1,256] range is performed. In the last stage, coupling of the spatial and the temporal sources of information is achieved via a sparsity regularized Autoencoder, where the parameter for probability of neuron activation was set to 0.1, leading to sparse activations.

The features extracted by the VGG-16 network were subsequently introduced for training with three kernel-based SVM classifiers, namely, linear, Gaussian (with $\sigma = 1$) and 2^{nd} order

polynomial kernel SVMs. To generate the experimental results in the section subsection, we utilized the MatConvNet [25] platform and the available pre-trained models including the VGG-16, while for the fusion we employed the Matlab Neural Network toolbox which allowed the execution of the training process on an NVidia K2200 GPU. Performance is reported in term of classification error, *i.e.*, percentage of mis-classified sequences.

Detection using spatial features

In the first section of the experimental results, we investigate the case where 1000-dimensional spatial feature vectors are extracted for each frame. For each sequence, we perform temporal subsampling to 10 and 20 frames and as a result, each sequence is represented using either a 10000 or a 20000 dimensional feature vector. These spatial feature vectors are subsequently introduced to the SVM classifier for training or testing. The key drive for the experiment is to evaluate the impact of number of frames on the classification accuracy and we report the average of 10 realizations.

Figure 3 presents the classification error as a function of training examples using features extracted from 10 frames per sequence, while Figure 4 considers 20 frames per sequence. Regarding the impact of the number of frames, the results clearly indicate that using larger number of frames leads to significantly better and more stable performance. For the linear kernel SVM, we observe that using all the available training examples results in 22% error using 10 frames, as opposed to 16% when using 20 frames. As a far as the performance of each classifier is concerned, results also indicate that the Gaussian and the Polynomial kernel SVMs achieve the best performance, significantly better compared to the linear kernel case. Based on these finding, we fix the number of frames that are used for feature extraction to 20 in the rest of the reported experiments.

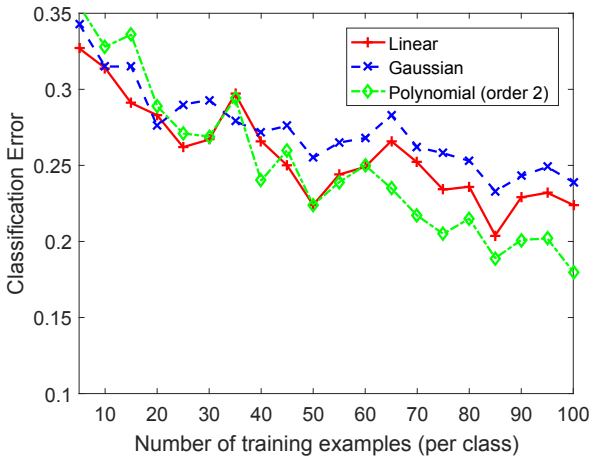

Figure 3: Classification error as a function of training examples using **10** frames per sequence, where only **spatial** features extracted by the VGG-16 network for three types of SVM kernels.

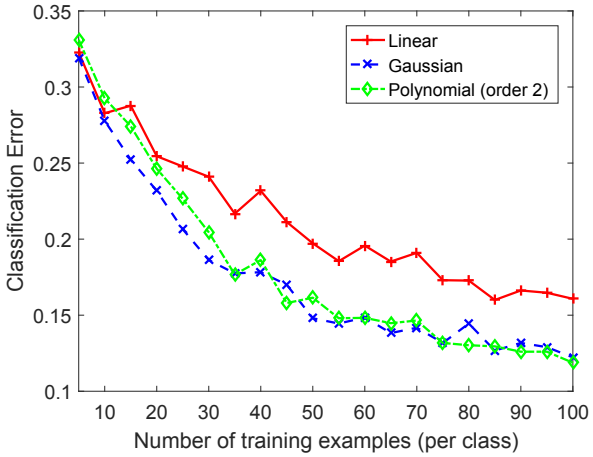

Figure 4: Classification error as a function of training examples using **20** frames per sequence and **spatial** features extracted by the VGG-16 network for three SVM kernels.

Detection using spatial and temporal features

We examine next the impact of temporal information when it is encoded through optical flow. Results are shown in Figure 5 for the case of independently introducing spatial and temporal features in the SVM classifier. Comparing the performance between the two case of inputs, spatial and spatio-temporal, experimental results suggest that temporal information can have a positive impact on detection accuracy. In terms of classifiers, we observe that non-linear kernels like the Gaussian and the Polynomial attain significantly lower classification error, albeit at the cost of extract computational requirements.

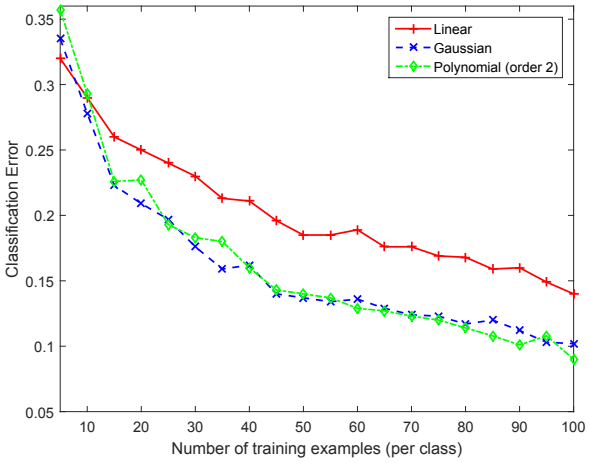

Figure 5: Classification error as a function of training examples, using **spatial** and **temporal** features extracted by the VGG-16 network for three types of SVM kernels.

Detection using spatio-temporal features

In the last section of the experimental validation we present the performance of event detection using fused spatio-temporal

features. Specifically, the outputs of the independently applied CNNs are concatenated and introduced to a regularized Autoencoder for feature fusion. The Autoencoder is tasked with generating compact representations, encoding both sources of information, and specifically, reducing the 40000-dimensional input vectors to 400-dimensional ones, two orders of magnitude smaller compared to the ambient dimensionality of the spatial and temporal feature.

Table 1: Classification error for different combination of classification kernel and features.

Method	Linear	Gaussian	Polynomial
Spatial	0.16	0.12	0.11
Spat. & Temp.	**0.14**	0.10	0.09
Fused	0.45	**0.02**	**0.03**

The results shown in Table 1 report the classification error achieved by all methods, including the case of spatio-temporal features, using all available (100) training examples per class. These results clearly demonstrate the huge potential which can be capitalized by fusion of different sources of information. Particularly, the two most powerful classifiers, the Gaussian and the Polynomial kernel SVM, are able to reduce their classification error almost an order to magnitude, even though the same information is used as input. The Gaussian Kernel especially is able to achieve an impressive 98% classification accuracy using only 100 training examples per class. An additional benefit of the fusion process is the dramatically smaller dimensionality of features extracted per sequences, reducing the requirements and facilitating subsequent analysis like classification.

To illustrate the potential of feature fusion for video analytics, Figure 6 presents the evolution of the value of the Autoencoder cost function during the iterative training process. We observe that that value is monotonically decreasing suggesting that better modeling of the data is achieved with more epochs. In this paper, we limited the maximum number of epochs to 1000 for reason of computational complexity, although better performance could potential be achieved with larger number of epochs.

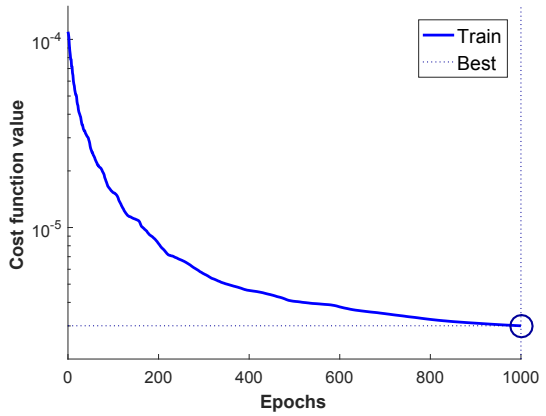

Figure 6: Evolution of the value of the cost function as a function of epoch number.

Conclusions

In this work, we focused our efforts in the design of a learning based architecture for the detection of events (goals) in sports (football) videos. We employ pre-trained convolutional neural networks for extracting both spatial and temporal features and we investigate Autoencoders for fusion of the different information sources. To validate the capabilities of the proposed architectures, a new dataset is compiled while experimental results suggest that very low classification error can be achieved, even from a limited number of training examples. In terms of computational complexity, when pre-trained CNN are utilized, the most demanding operation is the extraction of the optical flow information. Future work will examine the impact of deep stacked Autoencoders for achieving even higher classification performance.

Acknowledgments

This work was partially founded by the DEDALE project (contract no. 665044) within the H2020 Framework of the EC.

References

[1] M. Jefferson Ryan, and A. Savakis. "Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks." arXiv preprint arXiv:1612.00390, 2016.

[2] X. Zhongwen, Y. Yang, and A. G. Hauptmann. "A discriminative CNN video representation for event detection." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1798-1807, 2015.

[3] Y.-H. Ng, J. M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. "Beyond short snippets: Deep networks for video classification." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4694-4702, 2015.

[4] P. Xiaojiang, L. Wang, X. Wang, and Y. Qiao. "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice." Computer Vision and Image Understanding, 2016.

[5] Z. Shichao, Y. Liu, Y. Han, and R. Hong. "Pooling the Convolutional Layers in Deep ConvNets for Action Recognition." arXiv preprint arXiv:1511.02126, 2015.

[6] O. Ram Manohar, S. Sah, S. Pillai, and R. Ptucha. "Image description through fusion based recurrent multi-modal learning." In IEEE International Conference on Image Processing (ICIP), pp. 3613-3617, 2016.

[7] B. Hakan, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. "Dynamic image networks for action recognition." In IEEE International Conference on Computer Vision and Pattern Recognition CVPR. 2016.

[8] S. Karen, and A. Zisserman. "Two-stream convolutional networks for action recognition in videos." In Advances in Neural Information Processing Systems, pp. 568-576, 2014.

[9] S. Karen, and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, (2014).

[10] J. Yu-Gang, S. Bhattacharya, S.-F. Chang, and M. Shah. "High-level event recognition in unconstrained videos." International Journal of Multimedia Information Retrieval 2, no. 2, pp. 73-101, 2013.

[11] A.-H. Sami, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. "Youtube-8m: A large-scale video classification benchmark." arXiv preprint arXiv:1609.08675, 2016.

[12] T. Du, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. "Learning spatiotemporal features with 3d convolutional networks." In 2015

IEEE International Conference on Computer Vision (ICCV), pp. 4489-4497, 2015.

[13] M. Shugao, L. Sigal, and S. Sclaroff. "Learning activity progression in lstms for activity detection and early detection." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1942-1950, 2016.

[14] Y. Li, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. "Describing videos by exploiting temporal structure." In Proceedings of the IEEE International Conference on Computer Vision, pp. 4507-4515, 2015.

[15] K. Karalas, G. Tsagkatakis, M. Zervakis, and P. Tsakalides. "Deep learning for multi-label land cover classification." In SPIE Remote Sensing, International Society for Optics and Photonics, 2015.

[16] K. Alex, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105, 2012.

[17] G. Ross, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2014.

[18] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning." Nature 521, no. 7553, pp. 436-444, 2015.

[19] Y. LeCun, L. Bottou, G.B Orr, K. R. Muller. "Efficient backprop." In Neural networks: Tricks of the trade (pp. 9-48). Springer Berlin Heidelberg, 2012.

[20] K. Fotiadou, G. Tsagkatakis, and P. Tsakalides. "Low light image enhancement via sparse representations." In Image Analysis and Recognition, Springer International Publishing, pp. 84-93, 2014.

[21] M. Alireza, and B. Frey. "k-Sparse Autoencoders." arXiv preprint arXiv:1312.5663, 2013.

[22] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. "Towards good practices for very deep two-stream convnets." arXiv preprint arXiv:1507.02159, 2015.

[23] C. Feichtenhofer, A. Pinz, and A. Zisserman. "Convolutional Two-Stream Network Fusion for Video Action Recognition." arXiv preprint arXiv:1604.06573, 2016.

[24] A. Diba, A.M. Pazandeh, and L. Van Gool, "Efficient Two-Stream Motion and Appearance 3D CNNs for Video Classification." arXiv preprint arXiv:1608.08851, 2016.

[25] A. Vedaldi, and L. Karel. "Matconvnet: Convolutional neural networks for matlab." In Proceedings of the 23rd ACM international conference on Multimedia, pp. 689-692, 2015.

Author Biography

Dr. Grigorios Tsagkatakis (g. male)received his Diploma and M.S. degrees in Electronics and Computer Engineering from the Technical University of Crete, Greece in 2005 and 2007 respectively, and his Ph.D. in Imaging Science from the Rochester Institute of Technology (RIT), New York, in 2011. During the period 2011-2013 he was a Marie Curie post-doctoral fellow at the Institute of Computer Science (ICS), FORTH and he is currently working as a research associate with the Signal Processing Laboratory at FORTH-ICS. He has co-authored more than 30 peer-reviewed conferences, journals and book chapters in the areas of signal and image processing, computer vision and machine learning.

Dr. Mustafa Jaber is a computer vision engineer at NantVision, Inc. in Culver City, California where he performs research and development work in the area of deep learning and machine vision with emphasis to mobile devices. He received his BS in electrical engineering from the Islamic University of Gaza, Gaza, Palestine, in 2003, and MS in the same discipline from the Rochester Institute of Technology (RIT), Rochester, New York, in 2007. He also received his PhD in imaging science from the Chester F. Carlson Center for Imaging Science at RIT in 2012. His research interests are in the areas of digital image understanding, visual search, image ranking, and near duplicate image detection. Dr. Jaber has several conference and journal publications and was named on five patent applications. In addition, Dr. Jaber is the main shareholder of Capture Caption app.

Prof. Panagiotis Tsakalides received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1995. He is a Professor of Computer Science at the University of Crete, and the Head of the Signal Processing Lab at the Institute of Computer Science, Foundation for Research and Technology-Hellas, Greece. His research interests lie in the field of statistical signal processing with emphasis in non-Gaussian estimation and detection theory, and applications in sensor networks, imaging, and multimedia systems.