DISCRIMINANT ANALYSIS REGULARIZATION IN LIGHTWEIGHT DEEP CNN MODELS

Maria Tzelepi   *Anastasios Tefas*

Department of Informatics, Aristotle University of Thessaloniki

ABSTRACT

In this paper, we first propose lightweight deep CNN models, capable of effectively operating on-drone, in order to address various classification problems, i.e. crowd, football player, and bicycle detection, in the context of media coverage of specific sport events by drones with increased decisional autonomy. Subsequently, we propose a regularization technique, namely *Discriminant Analysis* regularization, aiming to enhance the generalization ability of the proposed models. The experimental evaluation validates the enhanced performance of the proposed regularizer.

Index Terms— Discriminant Analysis Regularization, Convolutional Neural Networks, Drones, Deep Learning.

1. INTRODUCTION

During the recent years, Deep learning algorithms, [1, 2], and principally the deep Convolutional Neural Networks (CNN) have been established as one of the most promising avenues of research in computer vision, providing outstanding results in a plethora of computer vision tasks, [3, 4, 5, 6, 7, 8]. The major reasons behind their success lie in the Graphics Processing Units (GPUs) computational power and affordability, as well as in the availability of large annotated datasets.

Over the few recent years *drones*, have powerfully emerged in the media and entertainment industry, being applied in a wide spectrum of applications, ranging from entertainment to visual surveillance, rescue within the context of natural disasters [9], and medical emergencies [10]. Their capability of capturing shots of inaccessible places, as well as spectacular aerial shots, gradually displaces prior practices in media production. A major issue associated with the rise of drones is the demand of developing efficient models for various computer vision tasks, capable of addressing the additional challenges of drone-captured images (that is, small object size, occlusion, etc.), and also capable of running on-drone, that is with limited processing power.

The objective of this work is to propose a regularization method in order to enhance the generalization ability of the lightweight models, proposed to address various tasks involved in the context of media coverage of certain sport events (i.e. football match, bicycle race) by drones. That is, we develop lightweight models, capable of running on-drone, for crowd (addresing also the demand of safety), football player, and bicycle detection. Our goal is to provide semantic heatmaps by e.g. predicting for each location within the captured scene the crowd presence. That is, we train models with RGB input of size e.g. 128×128, and then high resolution test images are fed to the network, and for every window 128×128, we compute the output of the network at the last convolutional layer. We note that is of utmost importance for the drone to be able to handle high resolution images, since the objects in drone-captured scenes are of small size, and thus image resizing in order to render the deployment on-drone feasible, would further shrink them, making their detection even impossible. The above procedure finds also application in the camera control problem, [11], where the semantic heatmaps for each of the considered tasks, aim to assist the algorithm for controlling the camera of the drone for cinematography tasks by sending error signals. Subsequently, we propose a novel regularizer in order to control over-fitting and enhance the performance of the proposed models.

Generally, addressing the problem of over-fitting, which arises due to their large capacity, is a central issue associated with the deep neural models. During the past years, several regularization schemes have been proposed in order to prevent over-fitting in neural networks, ranging from common regularization methods, like L1/L2 regularization which penalize large weights during the network optimization, and early stopping of the training procedure, to Dropout [12] where for each training sample, a randomly selected subset of the activations is zeroed in each epoch, and a generalization of it, Dropconncet [13] which instead of activations, sets a randomly selected subset of weights within the network to zero. From a quite different viewpoint, multitask-learning [14] constitutes also a way of improving the generalization ability of a model. For example, in [15] the authors introduced techniques developed in semi-supervised learning in the deep learning domain. That is, they combined an unsupervised regularizer with a supervised learner to perform semi-supervised learning. In this work, we propose a new regularized training method, insipired by the Linear Discriminant Analysis (LDA) [16] algorithm, namely Discriminant Analysis (DA) regularization, which aims to enhance the discriminative power of the proposed models by forcing the training samples belonging to the same class to come closer to their class centroid.

The rest of the paper is organized as follows: In Section 2, we present the proposed regularization method. In Section 3 we provide the implementation details and the experimental evaluation of the proposed method, and finally, conclusions are drawn in Section 4.

2. PROPOSED METHOD

In this work, we propose a novel regularized training method, motivated by the LDA method, that aims at best separating training samples of different classes, by projecting them into a new lower dimensional space, that maximizes the between-class separability while minimizing their within-class variability. Specifically, in the proposed scheme, apart from the classification loss which preserves the between class separability, we introduce an additional regularization loss aiming to bring the training samples of the same class closer to the class centroid. Hence, in this way, the so-called DA regularizer enhances the discriminative power of the model.

Thus, for an input space $\mathcal{X} \subseteq \Re^d$ and an output space $\mathcal{F} \subseteq \Re^q$, we denote as $\phi(\cdot\,;\mathcal{W}) : \mathcal{X} \to \mathcal{F}$ a deep neural network with $N_L \in \mathbb{N}$ layers, and set of weights $\mathcal{W} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_{N_L}\}$, where $\boldsymbol{W}_l$ are the weights of a specific layer l. We also denote the set of weights up to layer l as $\mathcal{W}^l = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_l\}$. Then, the output of layer l for a given input $\boldsymbol{x}_i$ is computed as follows: $\phi(\boldsymbol{x}_i\,;\mathcal{W}^l) = \sigma_l(\boldsymbol{W}_l \cdot \phi(\boldsymbol{x}_i\,;\mathcal{W}^{l-1}) + \boldsymbol{b}_l)$, where $\sigma_l(\cdot)$ is the activation function of layer l, $\boldsymbol{b}_l$ the bias term, $\phi(\boldsymbol{x}_i\,;\mathcal{W}^{l-1})$ the output of the previous layer, and $\cdot$ denotes a linear operation (*e.g.* matrix multiplication or convolution). Hence, we consider a set $\mathcal{D}_N = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of training samples on $\mathcal{X}$, and their corresponding representations, $\phi(\boldsymbol{x}_i\,;\mathcal{W}^l)$, at the layer l. We also consider the set $\mathcal{Z}^i = \{\mathbf{x}_k, k = 1, \ldots, K^i\}$ of K^i samples belonging to the same class with the i-th sample.

Then, the objective of the DA regularizer is defined as follows:

$$\min_{\mathcal{W}^l} \mathcal{J}_{DA} = \min_{\mathcal{W}^l} \sum_{i=1}^{N} \|\phi(\boldsymbol{x}_i\,;\mathcal{W}^l) - \boldsymbol{\mu}_i\|_2^2, \qquad (1)$$

where $\boldsymbol{\mu}_i = \frac{1}{|\mathcal{Z}^i|} \sum_{x_j \in \mathcal{Z}^i} \phi(\boldsymbol{x}_j\,;\mathcal{W}^l)$.

Optimizing objective (1) lets the network learn parameters such that data samples belonging to the same class are closely mapped to their class centroid, enhancing the discriminative power of the model.

The proposed regularizer can be attached to one or multiple neural layers. Thus, for a deep neural model of N_L layers, the total regularization loss is formulated as: $L_{reg\_total} = \sum_{l=1}^{N_L} \lambda_l L_{reg_l}$, where L_{reg_l} is the regularization loss, as defined in (1), for a certain layer, l, while the parameter $\lambda_l \in [0, 1]$ controls the relative importance of the specific regularization loss. Then, the total loss in the regularized training scheme is computed by summing the classification loss and the total regularization loss. Either hinge loss or softmax loss can be utilized as classifiers. In our experiments we use the softmax classifier. We use gradient descent to solve the above optimization problem. It is, finally, noted that the proposed regularizer can be implemented over the entire dataset, for the centroids of all the samples belonging to one class, as well as in terms of mini-batch training. In our experiments we implement it in terms of mini-batch training.

3. EXPERIMENTS

In this section, we present the experiments performed in order to evaluate the proposed regularization method. Throughout this work, we use Test Accuracy (Classification Accuracy) to evaluate the proposed regularizer. Each experiment is repeated five times and we report the mean value and the standard deviation, considering the maximum value of Test Accuracy for each experiment. The probabilistic factor is the random weight initialization. The proposed CNN models serve as baseline for the proposed regularization method. We also compare the proposed regularizer with the common L1 and L2 regularizers. In the following, we describe the utilized CNN architecture, the utilized datasets, and the implementation details of the proposed method, and finally we present the validation results.

3.1. CNN models and Discussion on Speed

The proposed CNN model contains six convolutional layers. Since the input images of the utilized datasets are of various sizes (that is, 128×128, 64×64, and 32×32), we use appropriate pooling for each of the three cases. That is, for the first case, the network accepts RGB images of size $128 \times 128 \times 3$. The output of the last convolutional layer is fed to a softmax layer which produces a distribution over the 2 classes. Each convolutional layer except for the last one is followed by a Parametric Rectified Linear Unit (PReLU) activation layer which learns the parameters of the rectifiers, since it has been proven to enhance the classification results [17]. Max-pooling layers follow the first and the fifth convolutional layers, while a response-normalization layer is utilized after the first pooling layer. A Dropout layer [18] with probability 0.5 follows the fifth convolutional layer. An overview of the proposed model is illustrated in Fig. 1. In the second

Model	Input	Jetson TX2	GeForce GTX 1080
VGG	224 × 224	9.36	89.52
Proposed	224 × 224	49.7	416.66
Proposed	512 × 512	13.1	99.4
Proposed	1024 × 1024	2.1	23.45

Table 1: Speed (FPS)

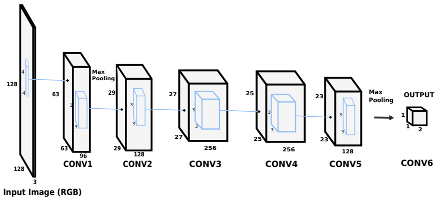


Fig. 1: Overview of the proposed CNN architecture

case, where the input size is 64 × 64, we remove the pooling layer which follows the 5^{th} convolutional layer, while in the third case where the inut size is 32 × 32, we also remove the first pooling layer from the initially described architecture.

We test the proposed model for the crowd detection task on a GeForce GTX 1080 GPU for various input sizes, and we compare it in terms of frames per second (FPS) with a common baseline model (i.e. VGG-16 [19]) for the latter's fixed input. Since the deployment of the detectors will be done on a drone, we also test the performance on a state of the art low-power GPU, that is an NVIDIA Jetson TX2 module with 8GB of memory. The results are presented in Table 1. As we can see, the proposed model operates at 49.7 fps for input of size 224 × 224, against the baseline model which runs at 9.36 fps for the same fixed input on the Jetson TX2 module, whereas it runs at 13.1 fps for input of size 512 × 512, and at 2.1 fps for input of size 1024 × 1024. We should also note that even if we discard the fully connected layers of the VGG model, and use only the fully-convolutional portion, the proposed model is considerably faster. For example, the modified fully convolutional VGG model runs at 28.16 fps for input 512 × 512 on the GTX 1080 (against 99.4 fps of the proposed one), while for an input of size 1024 × 1024 it is out of memory even in the GTX 1080.

3.2. Datasets

In order to evaluate the performance of the proposed DA regularizer we conduct experiments on three datasets, constructed for Crowd, Football Player, and Bicycle detection. The so-called Crowd-Drone dataset contains 11,840 train images of crowded scenes and non-crowded scenes. We use 2,368 images of them as test set. Input images are of size 128 × 128. The second dataset, constructed for football player detection consists of 98,000 train images of football players and non-football players, and a test set of 10,000 images. Input images are of size 32 × 32. Finally, the third dataset, namely Bicycles, contains 51,200 equally distributed train images of bicycles (bicycle with bicyclist) and non-bicycles, and a test set of 10,000 images. Input images are of size 64 × 64.

3.3. Implementation Details

The proposed CNN models were implemented using the Caffe Deep Learning framework [20]. The learning rate is set to 10^{-5}, and the batch size is set to 64. The weight decay is 0.0005, and the momentum is 0.9. All the models are trained on an NVIDIA GeForce GTX 1080 with 8GB of GPU memory, for 100 epochs.

As mentioned before, the proposed regularizer can be applied on individual layers, as well as on multiple layers. In our experiments, we apply the regularizer on all the convolutional layers. To do this, instead of using directly the high-dimensional features from a specific convolutional layer, we attach an additional pooling layer on each of these layers, namely Maximum Activations of Convolutions (MAC)[21] layer that implements the max-pooling operation over the height and width of the output volume, for each of the 128 feature maps of the CONV5 layer, correspondingly of the 256 feature maps of the CONV4, and so on. That is, the MAC layer, for example on CONV5 outputs a 128-d vector for each input image.

The regularization loss is initially significantly larger than the softmax one. Thus, in order to control the relative importance of the contributed losses, we first set the regularization loss parameter, λ, to 0.0001, and we fixed it to 0.01 at the 20 epochs up to the final epoch, for all the convolutional layers.

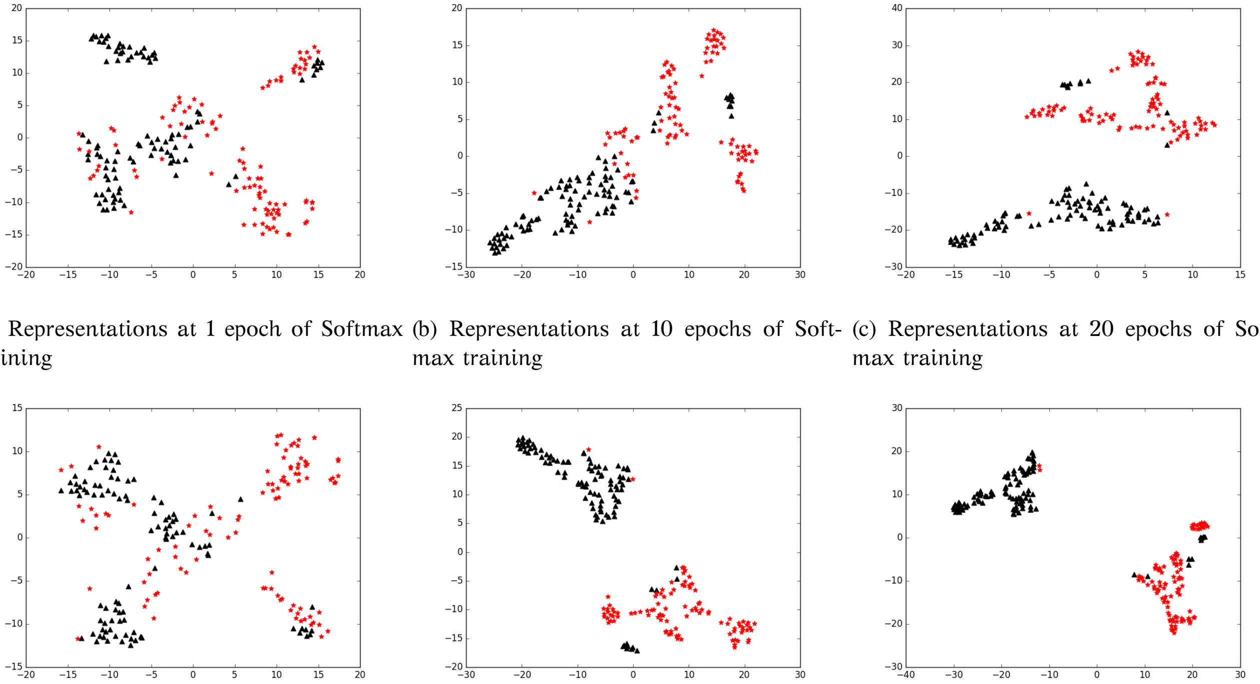
3.4. Experimental Results

In Table 2 we present the performance of the proposed regularizer against the softmax-only approach, in terms of Test Accuracy. We also compare the regularizer with the standard L1 and L2 regularization schemes. Best results are printed in bold. From the demonstrated results, we can see that the proposed regularizer considerably improves the classification performance, while it is also superior over the L1 and L2 regularizers, which either slightly improve the results or they harm the performance (e.g. L1 regularizer on Bicycles dataset).

Training Approach	Crowd-Drone	Football Player	Bicycles
Softmax	0.9405 ± 0.0079	0.8850 ± 0.0051	0.9119 ± 0.004
Softmax & L1	0.9435 ± 0.009	0.8834 ± 0.005	0.8991 ± 0.0079
Softmax & L2	0.9422 ± 0.005	0.8856 ± 0.0083	0.9134 ± 0.0021
Softmax & DA	**0.9546 ± 0.0061**	**0.9128 ± 0.003**	**0.9423 ± 0.0045**

Table 2: Test Accuracy

Furthermore, as we have previously mentioned, the hinge loss could also be utilized for the classification task,



(a) Representations at 1 epoch of Softmax training

(b) Representations at 10 epochs of Softmax training

(c) Representations at 20 epochs of Softmax training

(d) Representations at 1 epoch of DA training

(e) Representations at 10 epochs of DA training

(f) Representations at 20 epochs of DA training

Fig. 2: Visualization by t-SNE for the Crowd-Drone dataset

instead of the softmax classifier. To this aim, we also perform indicative experiments on the Crowd-Drone dataset using the hinge loss instead of the softmax one, and we apply the proposed DA regularizer. Thus, the only hinge loss training achieves Test Accuracy 0.9488 ± 0.0009, while the hinge loss with the DA regularizer 0.9584 ± 0.003. That is, we can indeed achieve improved results with the proposed regularizer using the hinge loss as the classification objective.

Subsequently, we use the t-distributed stochastic neighbor embedding (t-SNE) [22] algorithm, a non-parametric technique for dimensionality reduction, widely used for data visualization, to visualize the 128-d feature representations generated by the CONV5 layer of the proposed DA and the baseline softmax-only models, for 100 crowded and 100 non-crowded images. Thus, in 2a, 2b, and 2c of Fig. 2 we illustrate the 2-d t-SNE embedding of the CONV5 representations at 1 epoch, the t-SNE embedding of the representations at 10 epochs, and at 20 epochs of softmax-only training, respectively. In 2d, 2e, and 2f of the same figure we provide the corresponding DA representations. As we can observe, while the main sofmax classifier aims at separating the samples of different classes, the DA regularizer seeks to bring the training samples' representations of the same class together. That is, the proposed regularizer induces the training samples' representations to shrink, while also preserving discriminative power.

4. CONCLUSIONS

In this paper, we first proposed lightweight deep CNN models, for various recognition tasks, involved in the context of media coverage of specific sport events by multiple drones. That is, we proposed models for crowd, fooball player, and bicycle detection. Subsequently, we proposed a novel Discriminant Analysis regularization method, aiming to enhance the generalization ability of the proposed models. The experimental evaluation validates the effenctiveness of the proposed regularizer.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

5. REFERENCES

[1] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[2] Jurgen Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[3] Eleftherios Daskalakis, Maria Tzelepi, and Anastasios Tefas, "Learning deep spatiotemporal features for video captioning," *Pattern Recognition Letters*, vol. 116, pp. 143–149, 2018.

[4] Maria Tzelepi and Anastasios Tefas, "Deep convolutional image retrieval: A general framework," *Signal Processing: Image Communication*, vol. 63, pp. 30–43, 2018.

[5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[6] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1701–1708.

[7] Joseph Redmon and Ali Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[8] Maria Tzelepi and Anastasios Tefas, "Deep convolutional learning for content based image retrieval," *Neurocomputing*, vol. 275, pp. 2467–2478, 2018.

[9] Ludovic Apvrille, Tullio Tanzi, and Jean-Luc Dugelay, "Autonomous drones for assisting rescue services within the context of natural disasters," in *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*. IEEE, 2014, pp. 1–4.

[10] A Claesson, D Fredman, L Svensson, M Ringh, J Hollenberg, P Nordberg, M Rosenqvist, T Djarv, S Osterberg, J Lennartsson, et al., "Unmanned aerial vehicles (drones) in out-of-hospital-cardiac-arrest," *Scandinavian journal of trauma, resuscitation and emergency medicine*, vol. 24, no. 1, pp. 124, 2016.

[11] Nikolaos Passalis, Anastasios Tefas, and Ioannis Pitas, "Efficient camera control using 2d visual information for unmanned aerial vehicle-based cinematography," in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*. IEEE, 2018, pp. 1–5.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[13] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013, pp. 1058–1066.

[14] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[15] Jason Weston, Frédéric Ratle, and Ronan Collobert, "Deep learning via semi-supervised embedding," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1168–1175.

[16] Ronald A Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[18] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[19] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[21] Giorgos Tolias, Ronan Sicre, and Hervé Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *CoRR*, vol. abs/1511.05879, 2015.

[22] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.