Improving the Performance of Lightweight CNN models using Minimum Enclosing Ball Regularization

Maria Tzelepi and Anastasios Tefas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: {mtzelepi, tefas}@csd.auth.gr

***Abstract*—The aim of this paper is two-fold. First, we propose lightweight CNN models, capable of effectively operating on-drone for various classification problems, emerging in the context of media coverage of specific sport events by drones, i.e. crowd, football player, and bicycle detection. Subsequently, we propose a regularization method, namely *Minimum Enclosing Ball* regularization, in order to improve the generalization ability of the proposed models. The experimental evaluation on three datasets indicates the effectiveness of the proposed regularizer.**

***Index Terms*—Minimum Enclosing Ball Regularization, Convolutional Neural Networks, Drones, Deep Learning.**

1. Introduction

Over the recent few years *Drones*, have powerfully emerged in the media and entertainment industry. The application fields of Drones range from entertainment to visual surveillance, rescue within the context of natural disasters [1], and medical emergencies [2]. 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 settling the issue of 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.

Deep learning algorithms, [3], and especially the deep Convolutional Neural Networks (CNN) have been proven as one of the most effective avenues of research in computer vision, due to their outstanding performance in a plethora of computer vision tasks, [4], [5], [6], [7]. The major reasons underlying their success lie in the Graphics Processing Units (GPUs) computational power and affordability, as well as in the availability of large annotated datasets.

In this work, we first propose lightweight CNN models, addresing various classification tasks involved in the context of media coverage of certain sport events by drones with increased decisional autonomy. That is, we develop lightweight models capable of running on-drone for crowd detection (addresing also the demand of safety), football player detection, and bicycle detection. Our goal is to provide semantic heatmaps by e.g. predicting for each location within the captured scene the crowd presence, [8]. 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 using a sliding window of size 128×128, we compute the output of the network at the last convolutional layer for each location. An example of a crowd heatmap is provided in Fig. 1. We note that is of pivotal importance for the drone to handle high resolution images, since as we have previously mentioned, the objects to be detected in drone-captured images 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, [9], where the semantic heatmaps for each of the aforementioned tasks, aim at assisting the algorithm for controlling the camera of the drone for cinematography tasks by sending error signals.

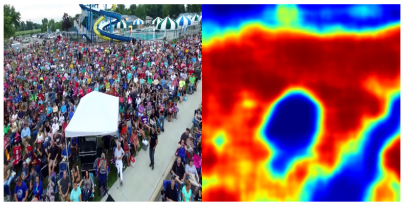

Figure 1: Crowded image and the corresponding predicted heatmap of crowd presence.

Subsequently, we propose a novel regularizer in order to control over-fitting and improve the performance of the proposed lightweight models. Addressing the problem of over-fitting, which arises due to their large capacity, constitutes in general a pivotal issue associated with the deep neural models. Over the past years, several regularization methods have been proposed in order to prevent over-fitting in neural networks, ranging from early stopping of the training procedure, or common regularization methods, like L1 and L2 regularization that penalize large weights during the network optimization, to Dropout [10] where for each training sample, a randomly selected subset of the activations is zeroed in each epoch, and a generalization of it, Dropconncet [11] which instead of activations, sets a randomly selected subset of weights within the network to zero. Other earlier works include weight elimination, [12], and Bayesian methods, [13]. 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 aim at improving the generalization ability of the proposed models, by proposing a novel regularizer, namely *Minimum Enclosing Ball* (MEB), motivated by the radius-margin based Support Vector Machines (SVM), [16], [17], [18]. That is, apart from the classification loss which aims at distinguishing the training samples belonging to different classes, we introduce an additional regularization loss aiming at shrinking the radius of the minimum enclosing ball of the training samples. The motivation behind the proposed method is that in binary classification problems there is one class and anything than this specific class (and correspondingly in multi-class problems over the one-versus-all approach), and thus the representations, especially of the negative class, may be extremely expanded in the feature space generated by the neural layer, as the classifier aims at distinguishing between the classes. Therefore, we propose to regularize the classifier by forcing the training samples' representations to come closer to their centroid. The proposed regularizer is generic and can be applied in several deep learning architectures for classification purposes.

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. Minimum Enclosing Ball Regularization

In this paper, we propose to improve the performance of the proposed lightweight models, by introducing a novel regularization objective with its motivational roots in the radius-margin based Support Vector Machines (SVM) [16], [17], [18]. Particularly, in [19], it is stated that the generalization error bound of the max-margin SVMs depends on not only the squared separating margin, γ^2, of the positive/negative training samples, but on the radius-margin ratio, R^2/γ^2, where R is the radius of the minimum enclosing ball of all the training samples. For a fixed feature space, the dependency of the error bound on the radius can be ignored in the optimization procedure, since the radius, R, is constant. However, when R is determined by the minimum enclosing ball of the training data, the model has the risk that the margin can be increased by simply expanding the minimum enclosing ball of the training data in the feature space. In order to remedy this problem, an algorithm that optimizes the error bound taking account of both the margin and the radius, in the context of Multiple Kernel Learning, is proposed in [17]. In [20], the authors also propose to incorporate a radius-margin bound as a regularization term into the classification loss of a deep model for $3D$ human activity recognition.

Towards this end, considering our binary classification tasks, as the softmax layer aims at separating the training samples' representations belonging to different classes, we propose to attach a regularization objective that aims at shrinking the radius of the minimum enclosing ball of the training samples, since representations, especially of the negative class, may be extremely expanded in the feature space generated by the neural layer.

Let $\mathcal{X} = \{\mathbf{X}_i, i = 1, \dots, N\}$ be the set of N training images, and $\mathcal{Y}^L = \{\mathbf{y}_i^L, i = 1, \dots, N\}$ be the set of N corresponding representations of the deep neural layer, L. We abbreviate as R_{MEB}, the radius of the minimum enclosing ball of all the training samples. The squared radius is formally expressed by the following equation:

$$R_{MEB}{}^2 = \min_{R, \boldsymbol{y}_0^L} R^2, \quad s.t. \quad \|\boldsymbol{y}_i^L - \boldsymbol{y}_0^L\|_2^2 \lesssim R^2, \quad \forall i, \tag{1}$$

where $\boldsymbol{y}_0^L$ is the centroid of all the training samples $\boldsymbol{y}_i^L$.

However, this definition suffers from a major shortcoming. That is, it can not be applied in terms of mini-batch training, since it requires the centroid of all the training data. In order to tackle this issue, we utilize an approximation of the above definition. We express the radius of the minimum enclosing ball of the training data, using the maximum pairwise distance over all pairs of training samples. That is:

$$\tilde{R}_{MEB}^2 = \max_{i,j} \|\boldsymbol{y}_i^L - \boldsymbol{y}_j^L\|_2^2 \tag{2}$$

In [18] the authors proved that the radius R_{MEB} is well approximated by $\tilde{R}_{MEB}$ with the following inequality:

$$\tilde{R}_{MEB} \leqslant R_{MEB} \leqslant \frac{1 + \sqrt{3}}{2} \tilde{R}_{MEB} \tag{3}$$

Thus, instead of minimizing the squared radius of the smallest sphere enclosing all the training samples, for simplicity we minimize the squared diameter that is defined by the maximum pairwise distance over all pairs of the training samples, since this does not affect the solution of the minimization problem, and following also the work in [18].

Subsequently, since the approximated radius is defined over all the pairs of training samples, we first formulate the following minimization problem utilizing the softmax function over the max operator which is non-smooth, as it is shown in [20] and then we further relax the approximated radius to make it suitable for mini-batch training:

$$\min_{\mathbf{y}_i^L \in \mathcal{Y}^{\mathcal{L}}} \mathcal{J}_{MEB} = \min_{\mathbf{y}_i^L \in \mathcal{Y}^{\mathcal{L}}} \sum_{i,j}^{N} k_{ij} ||\mathbf{y}_i^L - \mathbf{y}_j^L||_2^2, \tag{4}$$

where

$$k_{ij} = \frac{e^{a||\mathbf{y}_i^L - \mathbf{y}_j^L||_2^2}}{\sum_{i,j}^{N} e^{a||\mathbf{y}_i^L - \mathbf{y}_j^L||_2^2}} \tag{5}$$

measures the correlation of the two samples, while the parameter a controls the approximation degree to max operator. When a is infinite, the approximation is identical to the max operator, while when $a = 0$, $k_{ij} = \frac{1}{N^2}$. The relaxed definition of eq. (4) allows for defining the minimization objective in terms of mini-batch training, instead of the whole dataset. That is, for a set $\mathcal{B}$ of training samples' representations of a batch, eq. (4) becomes:

$$\min_{\mathbf{y}_i^L \in \mathcal{Y}^{\mathcal{L}}} \mathcal{J}_{MEB} = \min_{\mathbf{y}_i^L \in \mathcal{Y}^{\mathcal{L}}} \sum_{\mathbf{y}_i^L, \mathbf{y}_j^L \in \mathcal{B}} k_{ij} ||\mathbf{y}_i^L - \mathbf{y}_j^L||_2^2, \tag{6}$$

Thus, for $a = 0$, $k_{ij} = \frac{1}{|\mathcal{B}|^2}$, where $|\mathcal{B}|$ is the cardinality of set $\mathcal{B}$, it is straightforward to show that the minimization problem can be formulated as follows in terms of mini batch training:

$$\min_{\mathbf{y}_i^L \in \mathcal{Y}^{\mathcal{L}}} \mathcal{J}_{MEB} = \min_{\mathbf{y}_i^L \in \mathcal{Y}^{\mathcal{L}}} \sum_{\mathbf{y}_i^L \in \mathcal{B}} ||\mathbf{y}_i^L - \boldsymbol{\mu}||_2^2, \tag{7}$$

where $\boldsymbol{\mu} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{y}_j^L \in \mathcal{B}} \mathbf{y}_j^L$.

Either the softmax loss (cross entropy loss) or the hinge loss can be utilized for the classification task. In our experiments we use the softmax classifier. Thus, for a set of N input images $\mathcal{X} = \{\mathbf{X}_i, i = 1, \dots, N\}$ and their corresponding representations, $\mathcal{Y}^L = \{\mathbf{y}_i^L, i = 1, \dots, N\}$, the softmax loss is defined as:

$$L_s = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} l_{i,k} log(p_{i,k}), \tag{8}$$

where K is the number of classes, $l_{i,k} \in \{0, 1\}$ is a binary indicator that takes the value 1 if the class label k is the correct classification for the sample i, and $p_{i,k}$ is the predicted softmax probability the sample i to belong to the class k. 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 (7), 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, $L_{total} = L_s + L_{reg\_total}$. We use gradient descent to solve the above optimization problem. We should highlight that the proposed regularizer is generic, in the sense that it can be attached to any neural layer, of any deep architecture, and can be combined with various classification losses (e.g. softmax loss, hinge loss), since it is not incorporated as an additional term in a specific classification loss function.

We finally note that Support Vector Data Description method, [21], inspired by the Support Vector Classifier, proposes a MEB-like objective in the One Class Classification problem, as the main objective in order to find the outliers. However, the proposed regularizer, as mentioned previously, is rooted in the radius-margin based Support Vector Machines (SVM) [16], [17], [18], and proposes the MEB objective as a regularization in the main classification objective, in order to improve the generalization ability of the binary classifier.

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 first describe the utilized CNN architecture, and the utilized datasets, then we report 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 learned 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 × 128 × 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 [22]. 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 [23] with probability 0.5 follows the fifth convolutional layer aiming at reducing over-fitting. An overview of the proposed model is illustrated in Figure. 2. In the second case, where the input size is 64×64, we remove the pooling layer which follows the fifth convolutional layer, while in

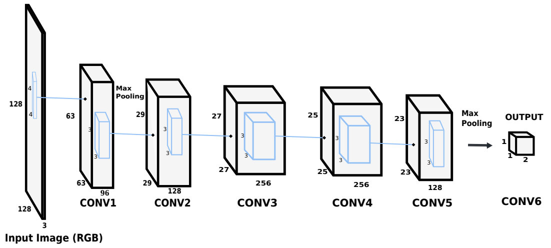

Figure 2: Overview of the proposed CNN architecture

the third case where the input 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 (the one with the larger input) 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 [24]) 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 used for on-board drone perception, 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 (the proposed model runs at 23.45 fps, respectively).

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)

3.2. Datasets

In order to evaluate the performance of the proposed MEB 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 [25]. 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) [26] 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).

Furthermore, as we have previously mentioned, the hinge loss could also be utilized for the classification task, 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 regularizer. Thus, the only hinge loss training achieves Test Accuracy 0.9488 ± 0.0009, while the hinge loss with the MEB regularizer 0.9541 ± 0.0022. That is, the proposed regularizer indeed exhibits superior performance over the baseline utilizing also the hinge loss as the classification objective.

Training Approach	Crowd-Drone	Bicycles	Football Player
Softmax	0.9405 ± 0.0079	0.9119 ± 0.004	0.8850 ± 0.0051
Softmax & L1	0.9435 ± 0.009	0.8991 ± 0.0079	0.8834 ± 0.005
Softmax & L2	0.9422 ± 0.005	0.9134 ± 0.0021	0.8856 ± 0.0083
Softmax & MEB	**0.9541 ± 0.0072**	**0.9448 ± 0.0057**	**0.9112 ± 0.01**

TABLE 2: Test Accuracy

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. Specifically, lightweight models for crowd, football player, and bicycle detection were proposed. Subsequently, we proposed a novel Minimum Enclosing Ball regularizer, aiming at enhancing the generalization ability of the proposed models. The experimental evaluation validates the effectiveness 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.

References

[1] L. Apvrille, T. Tanzi, and J.-L. 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.

[2] 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, p. 124, 2016.

[3] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[5] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

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

[7] E. Daskalakis, M. Tzelepi, and A. Tefas, "Learning deep spatiotemporal features for video captioning," *Pattern Recognition Letters*, vol. 116, pp. 143–149, 2018.

[8] M. Tzelepi and A. Tefas, "Human crowd detection for drone flight safety using convolutional neural networks," in *European Signal Processing Conference (EUSIPCO), Kos, Greece, 2017*.

[9] N. Passalis and A. Tefas, "Deep reinforcement learning for controlling frontal person close-up shooting," *Neurocomputing*, vol. 335, pp. 37–47, 2019.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. 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.

[11] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International Conference on Machine Learning*, 2013, pp. 1058–1066.

[12] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman, "Generalization by weight-elimination with application to forecasting," in *Advances in neural information processing systems*, 1991, pp. 875–882.

[13] D. J. MacKay, "Probable networks and plausible predictionsâ"a review of practical bayesian methods for supervised neural networks," *Network: Computation in Neural Systems*, vol. 6, no. 3, pp. 469–505, 1995.

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

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

[16] H. Do, A. Kalousis, and M. Hilario, "Feature weighting using margin and radius based error bound optimization in svms," *Machine Learning and Knowledge Discovery in Databases*, pp. 315–329, 2009.

[17] H. Do, A. Kalousis, A. Woznica, and M. Hilario, "Margin and radius based multiple kernel learning," *Machine Learning and Knowledge Discovery in Databases*, pp. 330–343, 2009.

[18] H. Do and A. Kalousis, "Convex formulations of radius-margin based support vector machines," in *International Conference on Machine Learning*, 2013, pp. 169–177.

[19] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[20] L. Lin, K. Wang, W. Zuo, M. Wang, J. Luo, and L. Zhang, "A deep structured model with radius-margin bound for 3d human activity recognition," *arXiv preprint arXiv:1512.01642*, 2015.

[21] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[22] K. He, X. Zhang, S. Ren, and J. 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.

[23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

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

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[26] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *CoRR*, vol. abs/1511.05879, 2015.