# Concept Detection and Face Pose Estimation Using Lightweight Convolutional Neural Networks for Steering Drone Video Shooting

Nikolaos Passalis and Anastasios Tefas
*Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece*
*passalis@csd.auth.gr, tefas@aiia.csd.auth.gr*

***Abstract*—Unmanned Aerial Vehicles, also known as drones, are becoming increasingly popular for video shooting tasks since they are capable of capturing spectacular aerial shots. Deep learning techniques, such as Convolutional Neural Networks (CNNs), can be utilized to assist various aspects of the flying and the shooting process allowing one human to operate one or more drones at once. However, using deep learning techniques on drones is not straightforward since computational power and memory constraints exist. In this work, a quantization-based method for learning lightweight convolutional networks is proposed. The ability of the proposed approach to significantly reduce the model size and increase both the feed-forward speed and the accuracy is demonstrated on two different drone-related tasks, i.e., human concept detection and face pose estimation.**

## 1. Introduction

*Unmanned Aerial Vehicles* (UAVs), also known as *drones*, are becoming increasingly popular for video shooting tasks since they are capable of capturing spectacular aerial shots that would be otherwise very difficult to obtain. However, flying a drone in a professional shooting setting requires at least two operators for each deployed drone, i.e., one for controlling the flight path and avoiding no-flight zones or other possible hazards, and one for controlling the camera.

Deep learning techniques, such as Convolutional Neural Networks (CNNs), can be utilized to assist various aspects of the flying and the shooting process allowing one human to operate one or more drones at once. This work is focused on developing techniques that will assist the video shooting of athletic events, e.g., road bicycle races or boat races. Several aspects of the flying and the shooting process can be assisted. First, a drone must be able to quickly identify whether one or more humans exist in a scene. Note that apart from the main drone camera, multiple smaller resolution cameras might be also available to aid this task. After humans have been identified in a scene, their positions are inferred and it is determined whether they are part of the crowd or they are persons of interest, e.g., the cyclists in the case of a bicycle race event. In the first case, the drone must fly away from the crowd (since the crowd defines a no-flight zone), while in the latter it must carefully follow the persons of interest, while avoiding any detected no-flight zones. After detecting a person of interest the drone must appropriately rotate the camera towards him. Since different shots have different specifications, the relative position of the person with respect to the camera, as well as the pose of his face (tilt and the pan) must be estimated in order to calculate the appropriate shooting angle.

Even though CNNs can be used to perform the aforementioned tasks, deploying them on drones is not straightforward, since there are significant memory and processing power constraints for any process running on-board. Therefore, it is crucial to develop lightweight deep learning models that would be able to run on-board and perform real-time classification and detection tasks. Existing state-of-the-art CNNs, such the VGG-16 [1], consist of hundreds of millions parameters, making them unsuitable for handling real-time tasks when the aforementioned constraints exist. Interestingly, most of the parameters of these models are not spent in the convolutional layers, but in the fully connected layers that are responsible for handling the feature maps extracted from the last convolutional layer. For example, for the VGG-16 network [1], almost 90% of the model's parameters are needed for just three fully connected layers, while only 10% of the parameters are used for its 13 convolutional layers. The reason for this is the large number of feature maps extracted from the last convolutional layer.

Motivated by the aforementioned observation a new type of quantization-based convolutional layer is proposed in this work to allow for reducing the size of CNN models. This layer is inspired by the well known Bag-of-Features (BoF) model [2], [3], which is used to quantize an arbitrary number of feature vectors into a constant length histogram representation. The pipeline of the Bag-of-Features model is the following. First, a number of feature vectors are extracted from an image that are then quantized into a predefined number of bins, that are called codewords. Finally, the histogram representation of each image is compiled by counting the number of feature vectors that were quantized into each bin. In this work the features extracted from the last convolutional layer are used. The proposed layer effectively quantizes and compresses the representation extracted from the convolutional layers, while also allows the network to handle arbitrary sized images. This is possible since the representation extracted from the proposed layer is decoupled from the size of the input image that is fed to the network. This is in contrast to regular CNN formulations where altering the input image size also alters the size of the extracted feature maps.

The main contribution of this work is the proposal of a novel BoF-based quantization layer that can significantly reduce the size of CNNs, while increasing their classification accuracy and speed. However, as explained in Section 3, the computational complexity of existing BoF formulations, such as [2], [3], is quite high thus making them unsuitable for the tasks considered in this work. To overcome this limitation, a fast linear variant of the BoF model is proposed. This variant can be readily implemented with existing deep learning tools allowing for easily using the proposed technique, even with existing pretrained CNNs. The proposed Convolutional Linear BoF model is evaluated using two different drone-related tasks, i.e., concept detection of humans using the VOC-2012 dataset [4], and pose (tilt and pan) estimation of human faces for calculating the optimal shooting angle [5]. It is experimentally demonstrated that the proposed method can greatly reduce the model size and increase the feed-forwarding speed, as well as improve the accuracy of the models.

The rest of the paper is structured as follows. Related work is introduced and compared to the proposed Convolutional Linear BoF model in Section 2. The proposed method is presented in detail in Section 3. Finally, the proposed method is evaluated in Section 4, while conclusions are drawn and future work is discussed in Section 5.

## 2. Related Work

Dealing with large CNNs is a well recognized problem in the literature. Many techniques have been proposed to reduce the model size [6], [7], [8], [9]. Usually compression and pruning techniques are used to reduce the size of CNN models [6], [7], [9]. Such techniques focus on compressing an already trained CNN, instead of training a CNN with fewer parameters in the first place. Some of these works [7], [9], also use vector quantization techniques. However, proposed the method uses a differentiable quantization scheme that allows for training both the quantizer and the rest of the network simultaneously. This is in contrast with compression techniques that merely use fixed quantization to reduce the size of the model. Nonetheless, the method proposed in this work can be combined with the aforementioned approaches to further reduce the size of the model.

The proposed method is also related to supervised dictionary learning approaches for the BoF representation [10], [11], [12], [13], [14]. All these methods are designed to work with handcrafted feature extractors instead of trainable convolutional layers. To the best of our knowledge, this is the first work that formulates the BoF model as a linear convolutional layer that can be used in any CNN network and allows for training the resulting network from scratch, while reducing the size of the model and increasing the feed-forwarding speed.

## 3. Proposed Method

The proposed Convolutional Linear BoF model, abbreviated as CL-BoF, is composed of a feature extractor, i.e., a set of convolutional and pooling layers, followed by the proposed Linear BoF layer and a set of fully connected layers. First, the feature extraction procedure is briefly discussed. Then, the proposed Linear BoF layer is presented in detail. Finally, the used fully connected architecture and the learning algorithm are presented.

Let $\mathbf{X}_i$ be an image that is fed to the convolutional feature extractor. Various CNN architectures, such as the VGG [1], or the ResNet [15], can be used for the feature extraction step after removing their fully connected layers. The last convolutional layer is used to extract the feature vectors that are fed to the proposed Linear BoF layer. The $j$-th feature vector extracted from the $i$-th image is denoted by $\mathbf{x}_{ij} \in \mathbb{R}^D$, where $D$ is the number of feature maps extracted from the last convolutional layer. The number of the extracted feature vectors is denoted by $N_i$ and depends on the size of the feature map and the used filter size.

After the feature extraction process, the $i$-th image is represented by a set of $N_i$ feature vectors $\mathbf{x}_{ij} \in \mathbb{R}^D$ $(j = 1, \dots, N_i)$. Instead of fusing the extracted feature vectors, which is the standard practice for CNNs [1], [15], the proposed Linear BoF layer is used to compile a fixed-length histogram for each image by quantizing its feature vectors into a predefined number of histogram bins/codewords. The length of the extracted histogram vector does not depend on the number of available feature vectors, allowing the network to handle images of arbitrary size without any modification and readily adapting to the available computational resources.

The Linear BoF layer is composed of two sublayers: an inner product layer that measures the similarity of the input features to the codebook and a pooling layer that builds the histogram of the quantized feature vectors. In the regular BoF model [3], the similarity between each feature vector $\mathbf{x}_{ij}$ and each codeword $\mathbf{v}_k$ is calculated as:

$$[\mathbf{d}_{ij}]_k = exp(\frac{-||\mathbf{v}_k - \mathbf{x}_{ij}||_2}{\sigma}) \in \mathbb{R} \tag{1}$$

where $\sigma$ is a scaling factor. However using this similarity definition requires calculating the pairwise distances between all the feature vectors and the codewords and then using the exponential function to transform these distances into similarities. Instead of using these computationally intensive steps, the Linear BoF uses the dot product to measure the similarity between each codeword and each feature vector:

$$[\mathbf{d}_{ij}]_k = abs(\mathbf{v}_k^T \mathbf{x}_{ij}) \in \mathbb{R} \tag{2}$$

where $abs(x)$ is the absolute value operator, which is used to ensure that Equation (2) properly encodes the similarity between the feature vectors and the codewords. This definition allows for a significant speed-up of the calculations. Furthermore, a regular convolutional layer can be used to implement the pairwise similarity calculation using $N_K$

filters of dimension $D \times 1 \times 1$, where $N_K$ is the number of the used codewords.

Then, $l^1$ normalization is used to obtain the normalized membership vector for each feature vector $\mathbf{x}_{ij}$:

$$\mathbf{u}_{ij} = \frac{\mathbf{d}_{ij}}{||\mathbf{d}_{ij}||_1} \in \mathbb{R}^{N_K} \tag{3}$$

This vector describes the similarity of the feature vector $\mathbf{x}_{ij}$ to each codeword $\mathbf{v}_k$. Finally, the histogram $\mathbf{s}_i$ is extracted for each image by pooling (using a sum pooling layer) the extracted membership vectors:

$$\mathbf{s}_i = \sum_{j=1}^{N_i} \mathbf{u}_{ij} \in \mathbb{R}^{N_K} \tag{4}$$

The Linear BoF layer receives the feature vectors of an image and compiles its histogram representation. This histogram is then fed to a classifier that decides the class of the image. To this end, a multilayer perceptron (MLP) with one hidden layer is used.

Without loss of generality it is assumed that the $i$-th training image is annotated by a label $t_i$ and there are $N_C$ different labels. It is straightforward to extend the proposed method to multi-label classification problems [16], or detection tasks [17].

Let $\mathbf{W}_H \in \mathbb{R}^{N_H \times N_K}$ be the hidden layer weights and $\mathbf{W}_O \in \mathbb{R}^{N_C \times N_H}$ be the output layer weights, where $N_H$ is the number of hidden neurons. Then, the hidden layer activations for the input histogram $\mathbf{s}_i$ of the $i$-th image are computed as:

$$\mathbf{h}_i = \phi^{(relu)}(\mathbf{W}_H \mathbf{s}_i + \mathbf{b}_H) \in \mathbb{R}^{N_H} \tag{5}$$

where $\phi^{(relu)}(x) = max(0, x)$ is the rectifier activation function which is applied element-wise and $\mathbf{b}_H \in \mathbb{R}^{N_H}$ is the hidden layer bias vector. The output of the MLP is calculated as:

$$\mathbf{y}_i = \phi^{(softmax)}(\mathbf{W}_O \mathbf{h}_i + \mathbf{b}_O) \in \mathbb{R}^{N_C} \tag{6}$$

where each output neuron corresponds to a label, $\mathbf{b}_O \in \mathbb{R}^{N_C}$ is the output layer bias vector and $\phi^{(softmax)}$ is the softmax activation function.

The categorical cross entropy loss is used for training the network:

$$L = -\sum_{i=1}^{N} \sum_{j=1}^{N_C} [\mathbf{t}_i]_j \log([\mathbf{y}_i]_j) \tag{7}$$

where $\mathbf{t}_i \in \mathbb{R}^{N_C}$ is the target output vector, which depends on the class of the input image ($t_i$) and it is defined as: $[\mathbf{t}_i]_j = 1$, if $j = t_i$, or $[\mathbf{t}_i]_j = 0$, otherwise.

All the parameters of the CL-BoF network can be learned using regular back-propagation and gradient descent:

$$\Delta(\mathbf{W}_{conv}, \mathbf{V}, \mathbf{W}_{MLP}) = -\eta(\frac{\partial L}{\partial \mathbf{W}_{conv}}, \frac{\partial L}{\partial \mathbf{V}}, \frac{\partial L}{\partial \mathbf{W}_{MLP}}) \tag{8}$$

where the notations $\mathbf{W}_{conv}$, $\mathbf{V}$, and $\mathbf{W}_{MLP}$ are used to refer to the parameters of the convolutional layers, the Linear BoF layer and the classification layers respectively. Instead of using the simple gradient descent, the Adam algorithm is utilized [18]. The Adam algorithm computes adaptive learning rates for each of the optimization parameters using estimates of the first and second moments of the gradient. For all the conducted experiments a learning rate of $\eta = 10^{-4}$ was used.

The codebook in the BoF model is usually initialized by clustering the set of all feature vectors $\mathcal{S} = \{\mathbf{x}_{ij} | i = 1, \ldots, N, j = 1, \ldots, N_i\}$, where $N$ is the number of training images, using the k-means algorithm. However, when the inner product similarity (Equation (2)) is used instead of the euclidean-based similarity (Equation (1)) it is no longer meaningful to use euclidean-based clustering. Indeed, it was experimentally established that randomly initializing the codebook (using Glorot uniform initialization [19]) works better than using k-means initialization. The fully connected layers were also initialized using Glorot uniform initialization.

## 4. Experiments

### 4.1. Evaluation Setups

The proposed method is evaluated on two different drone-related concept detection and classification tasks. The first task concerns concept detection [20], where the concept of *humans* is to be detected. As described before, being able to quickly detect humans in a given scene is crucial to appropriately control both the flight path and the camera of the drone. For the concept detection task, the VOC-2012 dataset, which contains images with and without humans, is used. The VOC-2012 is a challenging dataset, since humans appear in a wide range of natural settings. The predefined train split (5717 images) and validation split (5823 images) were used. Approximately 40% of the dataset images depict a human.

The second task is related to face pose estimation. After detecting the face of the main actor, e.g., a cyclist, the drone must be able to identify his pose to calculate the appropriate shooting angle. The Head Pose Image Database [5], is used to evaluate the proposed method for this task. Both the *pan* and the *tilt* of each face must be predicted. The Head Pose Image Database contains discrete annotations for the tilt (-90, -60, -30, -15, 0, +15, +30, +60, and +90 degrees) and the pan (-90, -75, -60, -45, -30, -15, 0, +15, +30, +45, +60, +75, and +90 degrees). Since the pose estimation procedure is expected to run after a face has been detected, the face was extracted from each image using the supplied annotations.

*Accuracy*, *average precision per class* (macro precision) and *average recall per class* (macro recall) were used to evaluate the classification performance. Accuracy measures the percentage of the predicated labels that exactly match the ground truth. Precision is defined as the ratio of true positives over the sum of true positives and false positives, while recall is defined as the ratio of true positives over the sum of true positives and false negatives. For the head pose

TABLE 1. CONCEPT DETECTION EVALUATION

| Method | # Conv. Params | # BoF Params | # FC Params | # Total Params | Accuracy | Recall | Precision |
|---|---|---|---|---|---|---|---|
| CNN | 6,529,664 | - | 4,719,362 | 11,249,026 | 78.81 | 77.95 | 77.58 |
| CL-BoF (8) | 6,529,664 | 4,096 | 706 | 6,534,466 | 81.13 | 77.26 | 82.64 |
| CL-BoF (128) | 6,529,664 | 65,536 | 33,538 | 6,628,738 | **82.43** | **78.25** | **85.16** |

TABLE 2. CONCEPT DETECTION: ACCURACY AND SPEED EVALUATION FOR IMAGES OF VARIOUS SIZES USING THE PROPOSED CL-BOF MODEL

| Image size | Clas. Accuracy | Clas. Time per Image |
|---|---|---|
| 224x224 | 82.43% | 9.11 msec |
| 200x200 | 82.62% | 6.91 msec |
| 176x176 | 82.12% | 5.10 msec |
| 152x152 | 81.69% | 3.68 msec |
| 128x128 | 79.13% | 2.52 msec |
| 104x104 | 74.65% | 1.65 msec |

estimation, the mean prediction error (in degrees) is also reported.

## 4.2. Concept Detection Evaluation

The concept detection evaluation results are shown in Table 1. For both the CNN and the CL-BoF models the convolutional layers were initialized using a VGG model [1], that was pretrained on ImageNet [21]. Batch normalization is used after the last convolutional layer [22]. For the baseline CNN model the initial fully connected layers were discarded and replaced by a 256×2 fully connected layer. For the CN-BoF model the fully connected layers were also replaced by a Linear BoF layer followed by a 256×2 fully connected layer. Either 8 codewords (CL-BoF (8)) or 128 codewords (CL-BoF (128)) were used for the Linear BoF layer. For the CL-BoF (8) a 64×2 fully connected layer was utilized. During the training process, the input images were randomly flipped and rotated (up to 10 degrees) with probability 0.5. The optimization ran for 10,000 iterations using batch size 16. The last pooling layer was removed from the CL-BoF model during the test to ensure that an adequate number of feature vectors are extracted regardless the used image size.

The proposed CL-BoF model performs better than the regular CNN model, even when only 8 codewords are used. Using more codewords, increases the detection accuracy even more, but also requires slightly more parameters. Even when 128 codewords are used the total parameters used after the convolutional layers are reduced by almost two orders of magnitude, while the total number of parameters in the network are reduced by 41%.

In the previously conducted experiments the input images were resized to 224×224 pixels. However, the CL-BoF is also capable of handling arbitrary sized images without re-training. This allows for readily adapting the model to the available computational resources (the time needed for feed-forwarding the network depends on the size of the input image). In Table 2, the classification accuracy and the classification speed is compared for images of various sizes. A mid-range GPU with 4GB of RAM was used for the conducted experiments. Feeding smaller images into the network can greatly reduce the classification time without significantly harming the classification accuracy. Note that this was not possible with regular CNN formulations, since altering the size of the input images also alters the size of the extracted feature maps (the fully connected layers can only process inputs of fixed size, in contrast to the proposed Linear BoF layer). The ability to process images of various sizes with lower computational cost is especially important in order to effectively use the secondary low-resolution cameras of a drone.

## 4.3. Head Pose Estimation Evaluation

For the head pose estimation evaluation the following convolutional architecture was used for extracting the feature vectors. Four convolutional layers (two with 32 $3 \times 3$ filters and another two with 64 3×3 filters), followed by a 2×2 max pooling layer and another four convolutional layers (two with 128 3×3 filters and another two with 256 3×3 filters) and a 2×2 max pooling layer were utilized. Again, batch normalization was used after the last convolutional layer and the rectifier activation function was utilized on all layers. Both the CNN and the CL-BoF networks were trained from scratch for 20,000 iterations using batch size of 32. Two sets of Linear BoF and fully connected layers were used (one set for tilt estimation and one set for the pose estimation). The same number of hidden neurons and codewords as in the concept detection experiments were used for each set. All the images that were fed to the network was resized to 48×48 pixels.

The experimental results are reported in Table 3. As before, using the proposed CL-BoF model reduces the pose estimation error by more than 25%, while allowing for using a smaller network (the size of the network is reduced over 75% compared to a regular CNN).

## 5. Conclusions and Future Work

In this paper a novel fast BoF-based quantization layer for training and deploying lightweight convolutional neural networks was proposed. This layer can reduce the size of CNNs, as well as increase the speed of the feed-forward process and the classification accuracy. The proposed Linear BoF can be readily implemented with existing deep learning tools allowing for easily using the proposed technique, even with existing pretrained CNNs. The proposed Convolutional

TABLE 3. FACE POSE (PAN AND TILT) EVALUATION

| Method | # Conv. Params | # BoF Params | # FC Params | # Total Params | Accuracy (Pan / Tilt) | Error (Pan / Tilt) |
|---|---|---|---|---|---|---|
| CNN | 1,172,768 | - | 4,724,758 | 5,897,526 | 68.92% / 61.94% | 6.68°/ 7.05° |
| CL-BoF (8) | 1,172,768 | 4,096 | 2,582 | 1,179,446 | 71.72% / 66.23% | 5.52°/ 5.55° |
| CL-BoF (128) | 1,172,768 | 65,536 | 71,702 | 1,310,006 | **75.38% / 67.10%** | **4.60°/ 5.15°** |

Linear BoF model was evaluated using two different drone-related tasks, i.e., concept detection of humans using the VOC-2012 dataset and pose (tilt and pan) estimation of human faces for calculating the optimal shooting angle. It was experimentally established that the proposed layer can significantly increase the classification/detection speed and accuracy, while reducing the total number of parameters needed for the model.

There are several interesting future work directions. First, the proposed method can be combined with convolutional object detectors, e.g., [17], to increase their speed and reduce the size of the deployed models. Also, spatial pyramid matching schemes similar to [23] can be used to further increase the accuracy of the model. Finally, the proposed approach can be extended to learn compact representations for various tasks, such as face recognition, using discriminant autoencoding techniques similar to [24].

## Acknowledgments

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

[2] J. Sivic, A. Zisserman *et al.*, "Video google: A text retrieval approach to object matching in videos." in *In Proceedings of the International Conference on Computer Vision*, vol. 2, no. 1470, 2003, pp. 1470–1477.

[3] N. Passalis and A. Tefas, "Neural bag-of-features learning," *Pattern Recognition*, vol. 64, pp. 277–294, 2017.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[5] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *FG Net Workshop on Visual Observation of Deictic Gestures*, 2004.

[6] M. Denil, B. Shakibi, L. Dinh, N. de Freitas *et al.*, "Predicting parameters in deep learning," in *In Proceedings of the Advances in Neural Information Processing Systems*, 2013, pp. 2148–2156.

[7] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," *arXiv preprint arXiv:1412.6115*, 2014.

[8] S. Han, H. Mao, and W. J. Dally, "A deep neural network compression pipeline: Pruning, quantization, huffman encoding," *arXiv preprint arXiv:1510.00149*, 2015.

[9] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," *arXiv preprint arXiv:1512.06473*, 2015.

[10] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1294–1309, 2009.

[11] X.-C. Lian, Z. Li, B.-L. Lu, and L. Zhang, "Max-margin dictionary learning for multiclass image categorization," in *In Proceedings of the European Conference on Computer Vision*, 2010, pp. 157–170.

[12] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, 2016.

[13] H. Lobel, R. Vidal, D. Mery, and A. Soto, "Joint dictionary and classifier learning for categorization of images using a max-margin framework," in *Image and Video Technology*, 2014, pp. 87–98.

[14] N. Passalis and A. Tefas, "Information clustering using manifold-based optimization of the bag-of-features representation," *IEEE Transactions on Cybernetics (in press)*, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[16] M.-L. Zhang, "Ml-rbf: Rbf neural networks for multi-label learning," *Neural Processing Letters*, vol. 29, no. 2, pp. 61–74, 2009.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010.

[20] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo, "Large-scale multimodal semantic concept detection for consumer video," in *In Proceedings of the International Workshop on Multimedia Information Retrieval*, 2007, pp. 255–264.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[23] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[24] P. Nousi and A. Tefas, "Deep learning algorithms for discriminant autoencoding," *Neurocomputing (in press)*, 2017.