Deep Multi-task Learning with Label Correlation Constraint for Video Concept Detection

Foteini Markatopoulou[1,2]
markatopoulou@iti.gr

Vasileios Mezaris[1]
bmezaris@iti.gr

Ioannis Patras[2]
i.patras@qmul.ac.uk

[1] Information Technologies Institute (ITI), CERTH, Thermi 57001, Greece
[2]Queen Mary University of London, Mile end Campus, UK, E14NS

ABSTRACT

In this work we propose a method that integrates multi-task learning (MTL) and deep learning. Our method appends a MTL-like loss to a deep convolutional neural network, in order to learn the relations between tasks together at the same time, and also incorporates the label correlations between pairs of tasks. We apply the proposed method on a transfer learning scenario, where our objective is to fine-tune the parameters of a network that has been originally trained on a large-scale image dataset for concept detection, so that it be applied on a target video dataset and a corresponding new set of target concepts. We evaluate the proposed method for the video concept detection problem on the TRECVID 2013 Semantic Indexing dataset. Our results show that the proposed algorithm leads to better concept-based video annotation than existing state-of-the-art methods.

Keywords

Concept detection; deep learning; video analysis

1. INTRODUCTION

Semantic concept detection in video refers to the task of assigning one or more semantic concepts to video fragments (e.g., video keyframes) based on a predefined concept list (e.g., “car”, “running”) [24]. In a typical process, the video is initially segmented into meaningful fragments, called shots; each shot may be represented by one or more characteristic keyframes; and, these keyframes are passed through a pre-trained deep convolutional neural network (DCNN) that performs the final class label prediction directly, using typically a softmax or a hinge loss layer [22, 10].

The small number of labeled training examples is a common problem in video datasets, making it difficult to train a deep network from scratch without over-fitting its parameters on the training set [23]. For this reason, it is common to use transfer learning. I.e., to take a network that has been trained on a large-scale source dataset (e.g., ImageNet [21]) and fine-tune its parameters for the target dataset. Furthermore, the tasks in the target dataset may be related, and so their relations can be exploited to further improve the video concept detection accuracy. Concept correlations obtained by the ground-truth annotation can provide a source of information regarding the relations between tasks. Additionally, concepts, besides label relations, can be related in terms of their feature representation or the task parameters, i.e., the parameters of the binary classifier learned from the training data. Multi-task learning (MTL) refers to those methods that learn many tasks together at the same time.

In this work we append a MTL-like loss to a neural network and we minimize the entire network end-to-end. In addition, we incorporate a label-based constraint related to the concept correlations. We refer to the proposed method as deep multi-task learning with label constraint (DMTL_LC) and we apply it on a transfer learning scenario. Specifically, we extend the two-sided neural network, proposed in [27] for MTL, in the following ways: i) We use the network jointly with a pre-trained network in order to perform transfer learning, instead of using it as a standalone network that takes as input hand-crafted or DCNN-based features. ii) We introduce a new label-based constraint that considers concept correlations. We evaluate DMTL_LC on the TRECVID 2013 semantic indexing (SIN) task's dataset of 38 different semantic concepts [20]. Our results show that the proposed algorithm leads to better concept-based annotation than existing state-of-the-art methods.

2. RELATED WORK

MTL and transfer learning are two strategies to improve learning by sharing knowledge across different but related tasks or domains. Let us define two domains with their learning tasks: the source domain D_{so} with a set of learning tasks T_{so} (the set of concepts that need to be detected), and the target domain D_{ta} with a set of learning tasks T_{ta}. On the one hand, transfer learning aims to improve the learning in D_{ta} by using the knowledge in D_{so}, without considering potential improvements to the tasks of D_{so}. The latter is the focus of multi-domain learning (MDL). On the other hand, MTL methods learn the relations across the learning tasks T_{so} or T_{ta} together at the same time. It should be noted that the terms MTL and MDL are sometimes used interchangeably. However, it is useful to distinguish them clearly: MDL refers to shared knowledge about the same tasks across different domains, while MTL refers to shared knowledge about different tasks in the same domain. The

MM '16, October 15-19, 2016, Amsterdam, Netherlands
© 2016 ACM. ISBN 978-1-4503-3603-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2964284.2967271

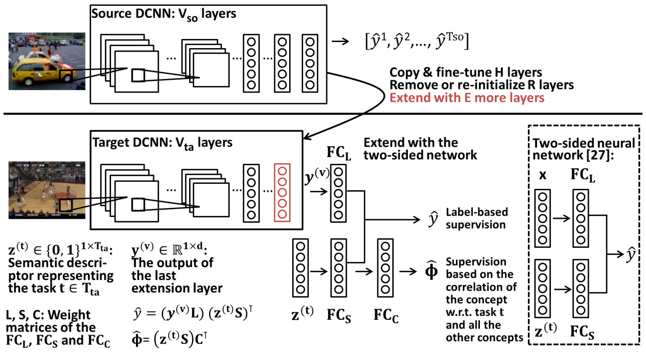
Source DCNN: V_{so} layers
$[\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^{Tso}]$
Copy & fine-tune H layers
Remove or re-initialize R layers
Extend with E more layers
Target DCNN: V_{ta} layers
FC_L
Extend with the two-sided network
$y^{(v)}$
$\hat{y}$
Label-based supervision
$\hat{\Phi}$
Supervision based on the correlation of the concept w.r.t. task t and all the other concepts
Two-sided neural network [27]:
x
FC_L
$\hat{y}$
$z^{(t)}$
FC_S
$z^{(t)}$
FC_S
FC_C
$z^{(t)} \in \{0,1\}^{1\times T_{ta}}$: Semantic descriptor representing the task $t \in T_{ta}$
$y^{(v)} \in \mathbb{R}^{1\times d}$: The output of the last extension layer
L, S, C: Weight matrices of the FC_L, FC_S and FC_C
$\hat{y} = (y^{(v)}L)(z^{(t)}S)^\intercal$
$\hat{\Phi} = (z^{(t)}S)C^\intercal$

Figure 1: Transfer learning using the proposed DMTL_LC method.

latter is the focus of this work, so we do not further discuss methods that focus on MDL, such as [12].

Noisy and incomplete annotations are common in video datasets (e.g., TRECVID SIN [20]), which makes it difficult to train a deep neural network from scratch [23]. Many works investigate which features within a pre-trained network are sufficiently generic, and develop approaches that effectively transfer this knowledge to new target datasets. The typical approach for transfer learning is to start with a DCNN trained in D_{so}, replace its classification layer with a new T_{ta}-dimensional classification layer and train it towards the D_{ta} domain [4, 29, 8]. The way that the parameters of the source DCNN will be used has been examined in many works. For example, in [18, 29], the first H layers of the pre-trained DCNN are copied and remain frozen, and the rest of the layers are randomly initialized. In addition, [29] fine-tunes the H layers, instead of freezing them, which leads to improved accuracy. Fine-tuning begins with the parameter weights of the source-domain DCNN and modifies them in order to adjust the network to the target domain. A different approach was proposed in [23, 15, 18] that extends a pre-trained DCNN by one or more fully-connected layers placed on the bottom of the classification layer.

MTL methods learn the relations across many tasks together at the same time. The main difference between MTL methods is the way they define task relatedness. Some methods identify shared features between different task and use regularization to model task relatedness [1, 17, 16]. Others identify a shared subspace over the task parameters [6, 5, 2]. The methods above make the strong assumption that all tasks are related; some newer methods consider the fact that some tasks may be unrelated. For example, the clustered MTL algorithm (CMTL) [31] uses a clustering approach to assign to the same cluster parameters of tasks that lie nearby in terms of their L2 distance. Adaptive MTL (AMTL) [25] decomposes the task parameters into a low-rank structure that captures task relations, and a group-sparse structure that detects outlier tasks. The GO-MTL algorithm [11] (i.e, for Grouping and Overlap in Multi-Task Learning) and the online version of it [14] use a dictionary-based method that allows two tasks from different groups to overlap by having one or more basis in common.

Deep learning is well suited for MTL; in [27] a two-sided neural network that addresses the MTL problem is proposed. Specifically, this method unifies several MTL methods that use a predictor matrix factorization approach, e.g., $\boldsymbol{w}^{(t)} = \boldsymbol{L}\boldsymbol{s}^{(t)^\intercal}$ [11], in order to learn their parameters using a two-sided neural network. $\boldsymbol{L}$ correspond to the parameter vectors of k latent tasks, while $\boldsymbol{s}^{(t)} \in \mathbb{R}^{1\times k}$ is a task-specific weight vector that contains the coefficients of the linear combination. MTL in deep learning architectures has also been proposed for facial landmark detection [30] and human pose estimation [19]. In [30] the task of facial landmark detection is optimized with the assistance of an arbitrary number of related/auxiliary tasks. This is a special case of the conventional MTL that typically maximizes the performance of all tasks. In this work the two sided neural-network proposed by [27] is modified and extended, for devising a deep learning method suitable for transferring a network that has been originally trained on a large-scale image dataset for concept detection, to a target video dataset and a corresponding new set of target concepts.

3. PROPOSED APPROACH

3.1 Problem Formulation

A video concept detection system needs to learn a number of supervised learning tasks T_{ta}, one for each target concept. Each task t is associated with the training set available for this concept $X^{(t)} = (\boldsymbol{x}_i^{(t)}, y_i^{(t)})_{i=1}^{N_t}$, where $\boldsymbol{x}_i^{(t)} \in \mathbb{R}^d, y_i^{(t)} \in \{\pm 1\}$. When the training set is small, it is common to take a DCNN that has been trained on a large-scale source dataset for T_{so} tasks, and transfer its parameters on a target DCNN to be trained on the target dataset $X = \{X^{(t)}\}_{t=1}^{T_{ta}}$ for a different set of T_{ta} tasks. With respect to the target dataset, the task parameters of related tasks may share similar knowledge, but also concept correlations obtained by the ground-truth annotation provide another source of information regarding the relations between tasks. In this section, considering all the above, we append a GO-MTL-like loss to a neural network and we incorporate a label-based constraint that considers concept correlations. We minimize the entire network end-to-end using stochastic gradient descent (SGD). We refer to the proposed method as deep multi-task learning with label constraint (DMTL_LC) and we apply it on a transfer learning scenario.

3.2 Deep Multi-task Learning with Label Constraint: DMTL_LC

Figure 1 presents the proposed approach for transferring a pre-trained DCNN network that consists of V_{so} layers (upper part) on a target DCNN to be trained to a target dataset (lower part). Starting with the DCNN trained on the source domain, the first H layers are copied to the target DCNN and fine-tuned on the target dataset. The remaining R layers are completely removed or randomly initialized; consequently, $H + R \leq V_{so}$. Subsequently, the target network can be extended with $E \geq 0$ fully-connected layers. Finally, the target network is trained using the DMTL_LC method.

The DMTL_LC algorithm unifies the GO-MTL algorithm [11] in the target DCNN by using and extending the two-sided neural network proposed in [27] as follows: i) The two-sided neural network is placed on the top of the V_{ta}-th fully-connected layer (where $V_{ta} = H + R + E$ is the number of layers before the two-sided network), instead of using it as a standalone network that takes as input hand-crafted or DCNN-based features. ii) The two-sided network is extended with a new label-based constraint in order to incorporate statistical information of pairwise correlations between concepts that we can acquire from the ground-truth annotation.

Specifically, the upper side of the target DCNN in Fig. 1, contains a fully-connected layer $FC_{\boldsymbol{L}}$ that takes as input the output of the V_{ta}-th layer. $FC_{\boldsymbol{L}}$ consists of k neurons, each representing one latent task. The parameter matrix $\boldsymbol{L} \in \mathbb{R}^{d \times k}$ of this layer constitutes a shared knowledge basis for all task models T_{ta}. The concept related to each task t is represented by a semantic descriptor $\boldsymbol{z}^{(t)} \in \{0,1\}^{1 \times T_{ta}}$, which is a binary vector of length T_{ta} that has zeros in every position except for position t. The lower side of the target DCNN contains a fully-connected layer $FC_{\boldsymbol{S}}$ that consists of k neurons and takes as input the semantic descriptor $\boldsymbol{z}^{(t)}$. Each row of the parameter matrix $\boldsymbol{S} \in \mathbb{R}^{T_{ta} \times k}$ of this layer contains a task-specific weight vector of the coefficients of the linear combination with the shared basis $\boldsymbol{L}$. This linear combination indicates for each concept which latent tasks describe it. The label-based constraint is placed on the top of the task-specific layer $FC_{\boldsymbol{S}}$. The network predicts a single output, which is equal to $\hat{y}^{(t)} = (\boldsymbol{y}^{(V_{ta})}\mathbf{L})(\boldsymbol{z}^{(t)}\mathbf{S})^{\mathsf{T}}$, where $\boldsymbol{y}^{(V_{ta})}$ is the output of the V_{ta} layer. The higher the output, the more likely that the concept learned w.r.t. task t is depicted in the input keyframe.

The above problem can be formulated by two separate objective functions:

$$\min_{(\mathbf{L},\mathbf{S},f\in F)} \frac{1}{T_{ta}} \sum_{t=1}^{T_{ta}} \left\{ \frac{1}{N_t} \sum_{i=1}^{N_t} \mathcal{L}\left(\hat{y}_i^{(t)}, y_i^{(t)}\right) \right\} \tag{1}$$

where $\hat{y}_i^{(t)} = (\boldsymbol{y}_i^{(v)}\mathbf{L})(\boldsymbol{z}^{(t)}\mathbf{S})^{\mathsf{T}}$ is the prediction w.r.t task t, and $\boldsymbol{y}_i^{(v)} = \alpha(\boldsymbol{W}^{(v)}\boldsymbol{y}_i^{(v-1)} + \boldsymbol{b}^{(v)})$ is the output of the v-th layer, with α referring to the layer's activation functions. E.g., $\alpha(x) = max(0,x)$ for the ReLU function.

In the above equation $\mathcal{L}$ refers to the loss function calculated between the prediction $\hat{y}_i^{(t)}$ and ground-truth annotation $y_i^{(t)}$. $f^{(v)} = \{\boldsymbol{W}^{(v)}, \boldsymbol{b}^{(v)}\}$ is the pair of the network parameters for the v-th layer and $F = \{f^{(v)}\}_{v=1}^{V_{ta}}$ is the set of network parameters for the first V_{ta} layers.

The second objective function that is placed on the top of the task-specific layer $FC_{\boldsymbol{S}}$ can be formulated as follows:

$$\min_{\mathbf{S}} \beta \Bigg(\frac{1}{T_{ta}} \sum_{t=1}^{T_{ta}} \left\{ \frac{1}{N_t} \sum_{i=1}^{N_t} \mathcal{L}\left(\hat{\boldsymbol{\phi}}^{(t)}, \boldsymbol{\phi}^{(t)}\right) \right\} \Bigg) \tag{2}$$

The role of this objective function is to approximate the correlation matrix $\boldsymbol{\Phi} \in [-1,1]^{T_{ta} \times T_{ta}}$. Each position of this matrix corresponds to the ϕ-correlation coefficient between two concepts regarding two different tasks t and t', calculated from the ground-truth annotation of the training set. Consequently, $\boldsymbol{\phi}^{(t)} \in [-1,1]^{1 \times T_{ta}}$ refers to the t'th row of $\boldsymbol{\Phi}$ that contains the correlations of task t with all the other tasks. $\hat{\boldsymbol{\phi}}^{(t)} \in \mathbb{R}^{1 \times T_{ta}}$, where $\hat{\boldsymbol{\phi}}^{(t)} = (\boldsymbol{z}^{(t)}\mathbf{S})\mathbf{C}^{\mathsf{T}}$, is the network's prediction for this row. Finally, $\boldsymbol{C} \in \mathbb{R}^{T_{ta} \times k}$ is the weight matrix to train for approximating the correlation matrix. To train $\boldsymbol{C}$, back propagation can be performed by the loss $\mathcal{L}$ between $\hat{\boldsymbol{\phi}}^{(t)}$ and $\boldsymbol{\phi}^{(t)}$.

We use the sigmoid cross entropy loss given by the following equation: $\mathcal{L} = \boldsymbol{\phi} log(\sigma(\hat{\boldsymbol{\phi}})) + (1-\boldsymbol{\phi}) log(1-\sigma(\hat{\boldsymbol{\phi}}))$, where $\sigma(.)$ refers to the sigmoid function $\sigma(x) = 1/(1+exp(-x))$. We scale the target vector $\boldsymbol{\phi}$ in [0,1] in order to deal with the negative values.

This second objective function takes the form of a constraint over the task-specific parameters $\mathbf{S}$ of the network. Specifically, the rows of the correlation matrix $\boldsymbol{\Phi}$ of two correlated concepts will be similar and we want the corresponding rows of $\boldsymbol{S}$ to be similar, too. During training, this second loss (Eq. 2) gets added to the total loss of the network (Eq. 1) with a discount weight β. At inference time, this auxiliary constraint is discarded.

4. EXPERIMENTS

4.1 Dataset and Experimental Setup

Our experiments were performed on the TRECVID 2013 SIN dataset [20], which consists of approximately 800 and 200 hours of internet archive videos for training and testing, respectively. The training set is partially annotated and highly imbalanced; approximately 100K positive keyframes are available for the 60 TRECVID SIN concepts, which is insufficient for training a DCNN from scratch. In our experiments, we used the 8-layer AlexNet [10] that was trained on 1000 ImageNet categories [21] as the source DCNN, and fine-tuned it on the 60 TRECVID SIN concepts. We evaluated all the methods on the test set using the subset of 38 concepts that were also evaluated as part of the TRECVID 2013 SIN task [20]. The video indexing problem was examined; that is, given a concept, we measure how well the top retrieved video shots for this concept truly relate to it. We analyze our results in terms of mean extended inferred average precision (MXinfAP) [28], which is an approximation of the mean average precision suitable for the partial ground-truth that accompanies the TRECVID dataset [20].

A first set of experiments was ran, where we examined different approaches of using the pre-trained AlexNet [10] to fine-tune a target-DCNN towards the 60 TRECVID SIN concepts [20]: i) The baseline approach (Fig. 1: H=7, R = E=0), that copies the first 7 layers [4, 29, 8]. ii) The extension approach (Fig. 1: H=7, R=0, E=1), that copies the first 7 layers and extends the network by one more layer [23, 15, 18]. iii) The re-initialization approach (Fig. 1: H=6, R=1, E=0), that copies the first 6 layers and randomly initializes the 7th layer [29]. For each approach we evaluated a) the typical transfer learning method (Default-TL) that replaces the classification layer of AlexNet [10] with a new 60-dimensional layer; b) the proposed DMTL_LC method that uses a two-sided network and considers concept correlations. The H layers, in all cases, were copied and fine-tuned towards the target dataset. To train the proposed method, for each concept, a training set was assembled that included all positive annotated training examples for the given concept, and negatives to a maximum of 15:1 ratio. For the Default-TL method we used the positive examples for each concept following an one-vs-all strategy. Subsequently, we applied each of the fine-tuned networks on the TRECVID keyframes and we evaluated the direct output of each network that corresponds to the class label prediction for 60 categories (Table 1).

We also compared (Table 2) the proposed method with the following ones: i) Single-task learning (STL) using a) Logistic regression (LR), b) LSVM and c) kernel SVM with radial kernel (KSVM). ii) MTL using: a) AMTL [25], b) CMTL [31] and c) the two-sided neural network instantiated with the GO-MTL algorithm [27]. We refer to the latter method as 2S-NN. STL refers to the typical approach of training one classifier e.g., SVMs, per concept, with features extracted from one or more layers of DCNNs [13, 4],

Table 1: MXinfAP (%) for 38 concepts, for different fine-tuning processes of the pre-trained 8-layer AlexNet [10] towards the 60 TRECVID SIN concepts [20]: i) The baseline [4, 29, 8]. ii) The extension [23, 15, 18]. iii) The re-initialization [29]. For each approach we evaluate a) the typical transfer learning method (Default-TL) that replaces the classification layer of AlexNet [10] with a new 60-dimensional layer; b) the proposed DMTL, DMTL_LC methods that use a two-sided network. The H AlexNet layers, in all cases, are copied and fine-tuned towards the target dataset.

fine-tuning process	(i) Baseline [4, 29, 8]	(ii) Extension [23, 15, 18]			(iii) Re-initialization [29]	
fine-tuning parameters	(a)	(b) #Neurons for the extension layer: 1096	(c) 2048	(d) 4096	(e) #Neurons for the re-initialization layer 1096	(f) 2048
DefaultTL-Softmax	**16.76**	16.22	15.53	14.79	16.24	16.68
DefaultTL-Hinge	13.26	19.91	19.89	18.76	19.20	15.30
Proposed-DMTL	12.71	15.82	14.89	19.93	18.39	19.47
Proposed-DMTL_LC	15.78	**20.13**	**22.60**	**20.84**	**22.54**	**21.47**

instead of performing the final class label prediction directly, using a softmax/hinge loss layer [22, 10]. To train the compared methods, we applied the pre-trained AlexNet on the TRECVID keyframes and we used as a feature the network's last fully-connected layer (fc8). Subsequently, we used the same training set of positive/negative examples as described above.

Regarding the proposed method, the value of k was set to 157 and the regularization parameter β in Eq. (2) was set to 0.3. These parameters are expected to depend on the dimensionality of the feature space and the number of examples, and according to preliminary experiments seem to work well for the employed dataset. The hinge loss was used in Eq. (1) and a ReLU function was placed on the top of $\boldsymbol{S}$ to encourage sparse models. We used stochastic gradient descent (SGD) with 0.9 momentum and cross-validated the learning rate between 10^{-5} and 10^{-2} by a multiplicative step-size $10^{0.5}$. The Caffe software [9] was used for training the DCNN networks on a Tesla K40 GPU. The LibLINEAR library [7] was used as the source of learning LSVM and LR models and the LibSVM [3] for learning KSVMs. The MALSAR library [32] was used for learning the CMTL [31] and AMTL [25].

4.2 Experimental Results

Tables 1 and 2 present the results of our experiments in terms of MXinfAP. DMTL is an intermediate version of the proposed DMTL_LC that solves the objective function of DMTL (eq. 1) without using the label constraint of DMTL_LC (eq. 2). In Table 1 we examine the best way of using the layers of the pre-trained AlexNet by comparing three different fine-tuning processes. For completeness, we also report how these processes affect the typical way of transferring learning that replaces the classification layer of AlexNet with a new 60-dimensional classification layer (Default-TL). Based on these results, which refer to the direct output of the fine-tuned networks, we can see that the proposed DMTL_LC performs better than the Default-TL alternative independently of the utilized fine-tuning process, with only one exception in the case of the baseline fine-tuning. Furthermore, adding the label constraint (DMTL_LC) further improves the DMTL method for all of the fine-tuning processes. The proposed DMTL_LC is overall the best performing method, reaching a MXinfAP of 22.60% (Table 1: col(c)). This result is important, considering that the pre-trained AlexNet was used as the source DCNN; by incorporating in our DMTL_LC framework better performing DCNN architectures such as GoogLeNet [26] instead of AlexNet, further performance gains are expected.

Table 2: MXinfAP (%) for 38 concepts for different STL and MTL methods using two pre-trained DCNNs.

	Methods	AlexNet	AlexNet Default-TL (best from Table 1)
	Direct output	-	19.91
STL e.g., [13, 4]	LR	18.57	22.34
	LSVM	20.59	22.21
	KSVM	18.81	21.79
MTL	AMTL [25]	20.44	22.21
	CMTL [31]	18.18	22.38
	2S-NN [27]	20.19	23.12
	Proposed DMTL_LC	**22.60**	**25.04**

In Table 2 we compare the proposed DMTL_LC method with different STL and MTL methods. The pre-trained AlexNet and the best Default-TL fine-tuned network of Table 1, i.e., Table 1: col. (b), are used as the source DCNNs. The proposed DMTL_LC fine-tunes each of these networks towards the 60 TRECVID SIN concepts. To train the other methods, the output of the last fully-connected layer of each source DCNN was used as a feature. Regarding the AlexNet source DCNN, the proposed DMTL_LC is the best performing method, reaching a MXinfAP of 22.60%. We also observe that fine-tuning is a procedure that significantly improves the precision of all the compared methods, by increasing the MXinfAP, when the best Default-TL is used as the source DCNN. As the Default-TL approach does not consider the correlations of the concepts and the relations across tasks, in contrast to the proposed DMTL_LC, the latter reaches the highest performance by fine-tuning once again the former network (MXinfAP equal to 25.04%).

5. CONCLUSIONS

In this work we presented deep multi-task learning method for video concept detection. Extensive experiments reveal the usefulness of fine-tuning a deep network by directly learning the relations between many task models (one per concept) in combination with the concept correlations that can be captured from the ground-truth annotation.

6. ACKNOWLEDGEMENTS

This work was supported by the EU's Horizon 2020 research and innovation programme under grant agreements H2020-687786 InVID and H2020-693092 MOVING, and by Nvidia corporation with the donation of a Tesla K40 GPU.

7. REFERENCES

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. *Advances in Neural Information Processing Systems (NIPS 2007)*, 2007.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[5] H. Daumé, III. Bayesian multitask learning with latent hierarchies. In *the 25th Conf. on Uncertainty in Artificial Intelligence (UAI 2009)*, pages 135–142, Quebec, Canada, 2009. AUAI Press.

[6] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2004)*, pages 109–117, Seattle, WA, 2004.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2014)*, 2014.

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[10] A. Krizhevsky, S. Ilya, and G. Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS 2012)*, pages 1097–1105, 2012.

[11] A. Kumar and H. Daume. Learning task grouping and overlap in multi-task learning. In *the 29th ACM Int. Conf. on Machine Learning (ICML 2012)*, pages 1383–1390, Edinburgh, Scotland, 2012.

[12] M. Long and J. Wang. Learning multiple tasks with deep relationship networks. *CoRR*, abs/1506.02117, 2015.

[13] F. Markatopoulou, V. Mezaris, and I. Patras. Cascade of classifiers based on binary, non-binary and deep convolutional network descriptors for video concept detection. In *the IEEE Int. Conf. on Image Processing (ICIP 2015)*, pages 1786–1790, Quebec, Canada, 2015.

[14] F. Markatopoulou, V. Mezaris, and I. Patras. Online Multi-Task Learning for Semantic Concept Detection in Video. In *the IEEE Int. Conf. on Image Processing (ICIP 2016)*, Phoenix, AZ, USA, 2016.

[15] F. Markatopoulou et al. ITI-CERTH in TRECVID 2015. In *TRECVID 2015*. NIST, USA, 2015.

[16] H. Mousavi, U. Srinivas, V. Monga, Y. Suo, M. Dao, and T. Tran. Multi-task image classification via collaborative, hierarchical spike-and-slab priors. In *the IEEE Int. Conf. on Image Processing (ICIP 2014)*, pages 4236–4240, Paris, France, 2014.

[17] G. Obozinski and B. Taskar. Multi-task feature selection. In *the 23rd Int. Conf. on Machine Learning (ICML 2006). Workshop of Structural Knowledge Transfer for Machine Learning*, Pittsburgh, PA, 2006.

[18] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2014)*, Columbus, OH, 2014.

[19] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2014)*, pages 2337–2344, Columbus, OH, 2014.

[20] P. Over et al. TRECVID 2013-An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID 2013*. NIST, USA, 2013.

[21] O. Russakovsky, J. Deng, and H. S. et al. ImageNet Large Scale Visual Recognition Challenge. *Int. Journal of Computer Vision (IJCV 2015)*, 115(3):211–252, 2015.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv technical report*, 2014.

[23] C. Snoek, D. Fontijne, K. E. van de Sande, and H. e. a. Stokman. Qualcomm research and University of Amsterdam at TRECVID 2015: Recognizing concepts, objects, and events in video. In *TRECVID 2015*. NIST, USA, 2015.

[24] C. G. M. Snoek and M. Worring. Concept-Based Video Retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322, 2009.

[25] G. Sun, Y. Chen, X. Liu, and E. Wu. Adaptive multi-task learning for fine-grained categorization. In *the IEEE Int. Conf. on Image Processing (ICIP 2015)*, pages 996–1000, Quebec, Canada, 2015.

[26] C. Szegedy et al. Going deeper with convolutions. In *the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*, Boston, MA, 2015.

[27] Y. Yang and T. M. Hospedales. A unified perspective on multi-domain and multi-task learning. In *the Int. Conf. on Learning Representations (ICLR 2015)*, San Diego, California, 2015.

[28] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *the 31st ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR 2008)*, pages 603–610, Singapore, 2008.

[29] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems (NIPS 2014)*, pages 3320–3328, 2014.

[30] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *the 13th Europ. Conf. on Computer Vision (ECCV 2014)*, pages 94–108, Zurich, Switzerland, 2014. Springer.

[31] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. *Advances in Neural Information Processing Systems (NIPS 2011)*, 2011.

[32] J. Zhou, J. Chen, and J. Ye. MALSAR: Multi-task learning via structural regularization. *Technical report*, 2011.