Neural Network Knowledge Transfer using Unsupervised Similarity Matching

Nikolaos Passalis, Anastasios Tefas
Department of Informatics, Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece
Email:passalis@csd.auth.gr, tefas@aiia.csd.auth.gr

***Abstract*—Transferring the knowledge from a large and complex neural network to a smaller and faster one allows for deploying more lightweight and accurate networks. In this paper, we propose a novel method that is capable of transferring the knowledge between any two layers of two neural networks by matching the similarity between the extracted representations. The proposed method is model-agnostic overcoming several limitations of existing knowledge transfer techniques, since the knowledge is transfered between layers that can have different architecture and no information about the complex model is required, apart from the output of the layers employed for the knowledge transfer. Three image datasets are used to demonstrate the effectiveness of the proposed approach, including a large-scale dataset for learning a light-weight model for facial pose estimation that can be directly deployed on devices with limited computational resources, such as embedded systems for drones.**

I. Introduction

Transferring the knowledge from a larger neural network to a smaller and faster one allows for deploying more lightweight and accurate networks. This technique is known as *neural network distillation* [1], *model compression* [2], or *knowledge transfer* [3]. Most of the proposed knowledge transfer techniques work as follows. A large neural network is used to generate soft-targets for each training sample. Then, these targets are used to train the lightweight model [1]. The *similarities* between the training samples are implicitly encoded in these soft-labels, providing more information regarding the training data than the plain binary labels. Also, using soft-labels instead of the ground truth labels regularizes the training process increasing the classification accuracy. In this paper, the larger model is also called *donor model*, while the smaller model is called *receiver model*.

Since most neural network knowledge transfer approaches follow the basic distillation idea, i.e., the donor is used to produce soft-labels using a *transfer* set of data and these labels are then used to train the receiver model, they are unable to transfer the knowledge between networks when the dimensionality of the layers used for the transfer is different. This limitation arises from the inability to provide meaningful similarity measures between vectors that have different dimensionality. Furthermore, this implies that it is not possible to directly use distillation to transfer the knowledge if the donor network predicts a larger number of classes than the receiver.

Several questions arise from the aforementioned observations. For example, is there a way to extract the knowledge that is encoded in a neural layer without merely regressing its output? Is it possible to transfer the knowledge between two layers of neural networks that differ in architecture? Note that existing knowledge transfer approaches usually employ a supervised term in the loss function in order to train useful networks [1]. Can we perform pure unsupervised knowledge transfer without using the knowledge transfer procedure merely as a regularizer?

The main contribution of this paper is the proposal of a method that is capable of transferring the knowledge between any two layers of two neural networks by matching the similarity between the training samples, effectively overcoming the limitations of the other knowledge transfer techniques. The geometry of the donor's feature space is sampled and then the receiver model is trained to mimic this geometry using similarity-induced embeddings [4]. To this end, the donor model is used to calculate the similarity matrix of the training set. The knowledge is transfered to the receiver by mimicking the similarities induced by the donor model, i.e., recreating the same geometry in a lower-dimensional space. It worths observing that the distillation approach [1], also tries to implicitly exploit such kind of similarity information by raising the softmax temperature and, thus, generating more fuzzy class assignments. Furthermore, note that the proposed method does not directly use the weights of the network or employ dimensionality reduction techniques to match the representation extracted from various parts of the network, as in [3], or [5]. Instead, it is the first *model-agnostic* method, that it is capable of *directly* performing knowledge transfer between any two layers of two networks, regardless the employed neural network architectures. Three image datasets are used to demonstrate the effectiveness of the proposed method. The performance of the proposed method is also evaluated for the problem of facial pose estimation using a light-weight model that can be deployed on embedded devices, such as drones that will assist the video shooting of sport events [6].

The rest of the paper is structured as follows. The related work is presented and compared to the proposed approach in Section II. Next, in Section III, the proposed method is presented in detail. Finally, in Section IV the proposed approach is thoroughly evaluated, while conclusions are drawn and possible future work is discussed in Section V.

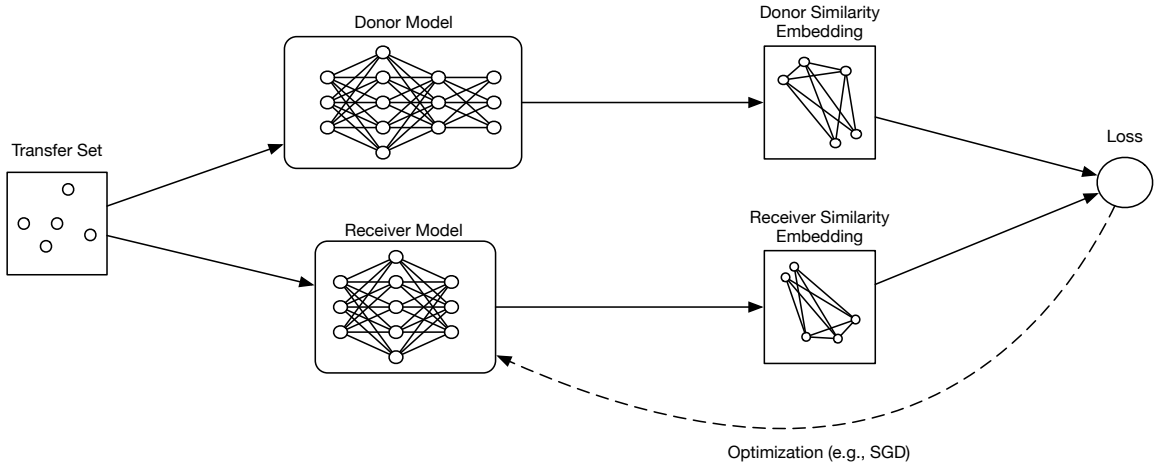

Fig. 1: Training the receiver model using similarity embeddings allows for transferring the knowledge from the donor model to the receiver model by "mimicking" the geometry of the donor model.

II. Related Work

Deep neural networks are becoming increasingly complexing fueling the research on transferring the knowledge of a large donor model into a smaller and faster one that can be deployed on embedded or mobile devices with limited computational power. Most of the proposed knowledge transfer methods use soft-labels, that are generated by the donor model, to train the receiver model [1], [2], [7], [8], [9]. Model compression [2], was among the first methods that used such kind of labels to perform knowledge transfer. This approach was extended in [1], by appropriately selecting the temperature of the softmax function before generating the soft-labels. The distillation approach also regularizes the training process providing better generalization than training the network directly using binary labels. In [9], the distillation process is used for domain adaptation using sparsely labeled data, while in [10] soft-targets are employed for pretraining a larger network.

The previous methods perform knowledge transfer by training the receiver model to mimic the soft-labels that were generated using the donor. A more direct approach is used in [3], where the weights of the donor model are used to initialize the receiver model increasing the convergence speed. The receiver model is then trained using the regular training dataset. A different approach is used in [5], where the distillation process is enriched using *hints* from the intermediate layers. Since the dimensionality of the intermediate layers is not always the same between different network, random projections are employed in [5], to match the dimensionality between the layers used for providing the hints.

To the best of our knowledge we propose the first method for knowledge transfer that works by matching the similarities induced by the donor and the receiver networks and thus being able to transfer the knowledge between any two layers, regardless the architecture of the corresponding networks. In contrast to the other approaches proposed in the literature, the proposed method does not use the weights of the donor network, as in [3], or employ random low dimension projections, as in [5]. Instead, the geometry of the feature space of the donor model is modeled using similarity embeddings, allowing for performing direct knowledge transfer from the donor model to the receiver model. Note that similarity embeddings are usually used to develop dimensionality reduction techniques, e.g., [4]. Nonetheless, as we demonstrate in this paper, they can be successfully used to transfer the knowledge between different neural networks.

III. Proposed Method

Let D denote the *donor* network from which the knowledge will be transfered to the *receiver* network R. To simplify the presentation of the proposed method, we assume that the networks receive a vector input. However, this is without loss of generality, since the method can readily work with any kind of input, e.g., tensors. The notations $D(\mathbf{x}, i)$ and $R(\mathbf{x}, i)$ are used to denote the output of the i-th layer of each network respectively, where $\mathbf{x} \in \mathbb{R}^L$ is an input vector (or tensor) and L is the input dimensionality. Note that we impose no constraint on the architecture of the models. For example, a $10 \times 500 \times 50 \times 20$ MLP can be used as the donor model, while a $10 \times 20 \times 5$ MLP can be used as the receiver model. The transfer set, which is employed to perform knowledge transfer, is denoted by $\mathcal{X}_{train} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and might contain any kind of data, e.g., the original training set, unlabeled data from a relevant domain or even synthetic data.

The proposed method aims to recreate the geometry induced by the the m-th layer of the donor network using the l-th layer of the receiver model. To achieve this, the geometry of the donor network is modeled using similarity embeddings and then the receiver model is trained to mimic these embeddings. The output of the donor's m-th layer is denoted by $\mathbf{t}_i = D(\mathbf{x}_i, m) \in \mathbb{R}^{L_m}$, while the output of the receiver's l-th layer is denoted by $\mathbf{y}_i = R(\mathbf{x}_i, l) \in \mathbb{R}^{L'_l}$. Note that the dimensionality of layers might different, i.e., $L_m \neq L'_l$,

allowing for transferring the knowledge between any two layers of the networks.

Let $[\mathbf{T}]_{ij}$ denote the similarity between the i-th and the j-th point of the transfer set, as expressed by the representation extracted from the donor model. Any similarity metric can be used to estimate this similarity, e.g., using the cosine similarity:

$$[\mathbf{T}]_{ij} = \frac{\mathbf{t}_i^T \mathbf{t}_j}{||\mathbf{t}_i||_2 ||\mathbf{t}_j||_2}, \quad (1)$$

or a Gaussian kernel:

$$[\mathbf{T}]_{ij} = exp(-\frac{||\mathbf{t}_i - \mathbf{t}_j||_2}{\sigma}). \quad (2)$$

To simplify the implementation and avoid the need for carefully estimating the scaling factor (σ) of the Gaussian kernel, a simpler linear kernel is used in this work:

$$[\mathbf{T}]_{ij} = |\mathbf{t}_i^T \mathbf{t}_j|. \quad (3)$$

The absolute value operator is employed to ensure that Eq. (3) is a proper similarity metric. Note that it has been recently demonstrated that replacing the Gaussian kernel with a simpler linear kernel can improve the speed of the methods without harming their accuracy. For example, soft formulations of the Bag-of-Features model, e.g., [11], usually employ a Gaussian kernel to calculate the similarity between the feature vectors and the codebook. However, replacing this kernel with a similar linear kernel does not significantly affect the quality of the learned representation [6]. The output of the donor network is normalized to the range $0 \dots 1$ using min-max scaling. Similarly, the similarity between the representations $\mathbf{y}_i$ and $\mathbf{y}_j$ extracted using the receiver model is defined as:

$$[\mathbf{P}]_{ij} = |\mathbf{y}_i^T \mathbf{y}_j|. \quad (4)$$

To train the receiver model to recreate the geometry of the donor the similarity matrices defined in Eq. (3) and in Eq. (4) must closely approximate each other. To this end, the receiver model is trained to minimize the mean squared error between the donor's similarity and the predicted similarity:

$$J = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} ([\mathbf{T}]_{ij} - [\mathbf{P}]_{ij})^2. \quad (5)$$

In this way, it possible to recreate the geometry of the donor model using the receiver model, as shown in Figure 1. This method allows for efficiently performing knowledge transfer, since the similarity between different samples expresses more information than a single binary label. However, the training process might rotate and distort the donor's feature space. Therefore, when the proposed method is employed to transfer the knowledge between the classification layers, a lightweight classifier, such as the nearest centroid classifier, must be used to recover the original class mapping.

The gradient descent method can be used to minimize the used loss function defined in Eq. (5):

$$\Delta \mathbf{W} = -\eta \frac{\partial J}{\partial \mathbf{W}}, \quad (6)$$

where the notation $\mathbf{W}$ is used to denote the parameters of the receiver model. In this work, the Adam algorithm [12], is used instead of the simple gradient descent since it provides faster and more stable convergence. Note that using the whole transfer set to calculate the loss function is computationally infeasible for large datasets, since a quadratic number of similarities must be calculated. Instead of using the whole training set for each optimization step, the optimization is performed by sampling a small number of transfer samples (batches) and calculating the corresponding similarity matrix. The data must be appropriately shuffled before each training epoch to ensure that different pairs of data points are employed for training the model in each epoch.

The proposed knowledge transfer algorithm is summarized in Figure 2. First, the receiver model is initialized (line 2) and the min-max scaler is initialized (line 3). Then, the transfer set is shuffled before each epoch (line 5), the output of the donor model is calculated (line 7) and the Adam algorithm is used to optimize the model according to the proposed loss function (Eq. (5)).

Input: The transfer set $\mathcal{X}_{train} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ and the donor model D

Hyper-parameters: N_{iters} (number of iterations), η (learning rate), N_{batch} (batch size) and the layers, m and l, used to transfer the knowledge (m-th layer of the donor model and l-th layer of the student model)

Output: The receiver model R

1: **procedure** KNOWLEDGE TRANSFER
2: Initialize the receiver model R
3: Calculate the minimum and the maximum values of the m-th layer of the donor model using the transfer set
4: **for** $i \leftarrow 1; i \leq N_{iters}; i++$ **do**
5: Shuffle the transfer set X_{train}
6: **for** $X_{batch} \in X_{train}$ **do**
7: Feed-forward the donor model and scale the representations $\mathbf{t}_i$ extracted from its m-th layer
8: Apply the Adam algorithm, using learning rate η, to update the parameters of the receiver model R in order to optimize Eq. (5).
return the model R

Fig. 2: Algorithm for knowledge transfer using similarity embeddings

IV. EXPERIMENTS

In this Section the proposed method is evaluated using three image datasets. First, the datasets used for the evaluation are introduced and the experimental setup is briefly described. Then, the experimental evaluation of the proposed method is presented in detail.

A. Datasets and Evaluation Setup

Three image datasets are used to evaluate the proposed method, the MNIST database of handwritten digits [13], the CIFAR10 [14], and the Annotated Facial Landmarks in the Wild (AFLW) dataset [15]. The MNIST database [13], is a well-known dataset that contains 60,000 training and 10,000 testing images of handwritten digits. There are 10 different classes, one for each digit (0 to 9), and the size of each image is 28×28. The CIFAR10 dataset [14], contains 60,000 32×32 color images that belong to 10 different categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. The dataset is already split into 50,000 train and 10,000 test images. The CIFAR10 dataset is a labeled subset of the 80 million tiny images dataset [16]. The Annotated Facial Landmarks in the Wild (AFLW) dataset [15], is a large-scale dataset for facial landmark localization. The AFLW dataset is used to evaluate the performance of the proposed knowledge transfer method for the problem of facial pose estimation using a light-weight model that can be deployed on embedded devices, such as drones that will assist the video shooting of sport events [6]. Estimating the facial pose of the actors allows for calculating the appropriate shooting angle according to the specifications of each shot. The pose estimation problem is expressed as a classification problem with three classes depending on the horizontal facial pose (yaw): left (yaw less that -10 degrees), center (yaw between -10 and 10 degrees) and right (yaw greater than 10 degrees). The 75% of dataset was employed for training the model and the rest 25% for evaluating the models. The supplied face annotations were used to localize and crop each face image. Each face image was then resized to 32×32 pixels, while images smaller than 16×16 pixels were discarded.

Note that it is not possible to directly evaluate the quality of the model after the knowledge transfer, since the embedded feature space may have been transformed, e.g., rotated. Therefore, two different approaches are used to evaluate the quality of the conducted transfer: a) a lightweight nearest centroid classifier was trained using only few training samples (3 samples per class) and b) the distillation method [1], was employed for finetuning the network towards classification. In the latter case, the output of the network can be directly used for predicting the class of a sample. Note that both proposed method and the distillation method are used in a purely unsupervised setting without using any labels during the training procedure.

B. MNIST Evaluation

First, we evaluate the proposed method using the well-known MNIST dataset. The following architecture was used for the donor model: two pairs of convolutional layers (64 filters of size 5×5) and 2×2 max pooling layers followed by two fully connected layers (512×10). For the receiver model a similar architecture is used, but with less filters leading to a network with over 30 times less parameters than the donor model (20k vs. 634k parameters). The receiver model is significantly simpler, using only 8 filters for the convolutional layers, while the size of the hidden layer is reduced to 128 neurons. For both networks rectifier activation functions are used. For training the baseline models dropout is also utilized for regularizing the training process. Training the donor model with the full training split of the MNIST dataset (denoted by "MNIST (F)") leads to a test error of 0.64%, while the receiver model achieves a test error of 1.17%. When the receiver network is trained using only 3 randomly sampled images per class, it achieves a classification test error of 28.33%. This dataset is denoted by "MNIST (30)". Note that the same setup was also used for training the nearest centroid classifier. When the full training dataset is used, then we train the models for 20 epochs, while when only 30 samples are used, then we use more epochs (20,000) to compensate for the smaller number of samples. The classification error for these baseline models and the number of parameters required for the donor (network A) and the receiver (network B) are summarized in Table I.

Network	# params	Data	Class. Error
A	634k	MNIST (F)	0.64%
B	20k	MNIST (F)	1.17%
B	20k	MNIST (30)	28.33%

TABLE I: MNIST: Donor model and receiver model baselines

The results are reported in Table II. The knowledge was transfered between the last convolutional layers of the networks and the optimization process ran for 50 epochs. We report two different classification error rates: the nearest centroid classifier error using the features extracted from the layer used for knowledge transfer ("NCC") and the classification error of the network when the output layer is directly used for classification ("NN"). The proposed approach is abbreviated as "KT" (Knowledge Transfer). Only the NCC error is reported for the KT method, since it was used to transfer the knowledge between the convolutional layers. We also compare the proposed method to the distillation approach [1] (left column of Table II). For the distillation approach, the final classification layer is used after raising the softmax temperature to $T = 10$. Again, the optimization for the distillation approach ran for 50 epochs. After transferring the knowledge using the proposed KT method, the network can be further finetuned using the distillation approach (both the NNC and the NN classification error rates are reported in this case).

For evaluating the knowledge transfer we used two different transfer sets. The first one uses a limited set of 30 training samples (abbreviated as "(30)"), while the second one uses the full training dataset (abbreviated as "(F)"). As it was expected, using only 30 samples for the knowledge transfer procedure leads to relatively large classification error. However, augmenting that data by adding Gaussian noise reduces the classification error by more than 6%. In both cases the proposed KT method significantly improves the learned representation over the distillation approach, e.g., the classification error for the NCC approach is reduced by more than 10% over the competitive methods. When the network is further finetuned

Transfer Set	Distillation		Knowledge Transfer	Knowledge Transfer + Distillation
	NCC	NN	NCC	NN
MNIST (30)	37.87%	27.10%	27.29%	27.21%	**26.77%**
MNIST (30) + $\mathcal{N}(0, 0.25)$	32.63%	25.48%	28.06%	27.57%	**20.23%**
MNIST (F)	17.83%	0.91%	12.29%	10.82%	**0.80%**

TABLE II: MNIST Evaluation: Knowledge transfer using different transfer sets (classification error rate)

Network	# params	Data	Class. Error
A	1,251k	CIFAR10 (F)	20.02%
B	422k	CIFAR10 (F)	28.31%
B	422k	CIFAR10 (30)	82.48%

TABLE III: CIFAR10: Donor model and receiver model baselines

with the distillation process, the NN classification error drops over 5% (MNIST (30) + $\mathcal{N}(0, 0.25)$), while when the full training dataset is used the proposed method reduces the NCC classification error by 7%, while reaching a spectacular 0.80% NN classification error. Note that the networks trained using the proposed method perform better than the baseline networks that were directly trained with the labeled dataset (Table I), indicating the practical value of the proposed method.

C. CIFAR10 Evaluation

Next, the CIFAR10 dataset is employed for evaluating proposed method. The following architecture is used for the donor model (network A): two convolutional layers with 32 filters of size 3×3, followed by one 2×2 max pooling layer, 2 convolutional layers with 64 filters of size 3×3 and a second 2×2 max pooling layer. Then, two fully connected layers (512×10) are used. The output of the convolutional layers is normalized using local response normalization [17], while dropout [18] is employed during the training process to ensure that the networks do not overfit the data. A simpler architecture is used for the receiver model (network B): two convolutional layer are used instead of four (the convolutional layers directly before the pooling layers are removed) and the size of the fully connected layer is reduced to 128×10. The donor and the receiver classification error (using both the full dataset "(F)" and a subsample of only 30 training samples "(30)") are reported in Table III. When the full training dataset is used the models are trained for 20 epochs, while when only 30 samples are used the optimization runs for more epochs (20,000) to compensate for the smaller number of samples. Note the classification accuracy of the receiver model is significantly reduced when trained using only 30 training samples.

As in the MNIST evaluation, the proposed KT method is first evaluated using only 30 training samples for the knowledge transfer, while the knowledge is transfered between the last convolutional layers of the networks. In this case, the proposed method achieves significantly better classification error than the baseline model, i.e., 75.57% vs. 82.48%. This improvement is mainly due to the use of the proposed KT transfer method, since using the distillation method alone achieves 80.28% classification error. Adding noise to the data further improves the classification error to 74.59%, while when the whole dataset is used for the knowledge transfer the classification error drops to 27.50% (combined with distillation-based finetuning), which outperforms the baseline model trained on the same data (28.31%). As before, the proposed knowledge transfer method improves the classification accuracy over directly training the models and outperforms the distillation approach.

D. AFLW Evaluation

Finally, the proposed method was also evaluated on the the AFLW dataset. The following architecture was used for the donor network: 2 convolutional layers with 16 3×3 filters, followed by a 2×2 max pooling layer, another 2 convolutional layers with 32 3×3 filters, a 2×2 max pooling layer and 128×3 fully connected layers. On the other hand, the receiver model consists of one 3×3 convolutional layer with 8 filters, a 3×3 max pooling layer, another one 3×3 convolutional layer with 16 filters, followed by a 3×3 max pooling layer and 16×3 fully connected layers. As before, local contrast normalization and rectifier activation functions are used. Note that the receiver model is significantly smaller than the donor model using almost two orders of magnitude less parameters, allowing the receiver model to be directly deployed on embedded devices with limited processing power. The networks were trained for 50,000 iterations using batches of 32 samples. The donor model achieved 87.19% pose estimation accuracy, while training the receiver model using the same setup leads to 82.83% accuracy.

Table V summarizes the experimental results using the ALFW dataset. For both the distillation and the KT methods 50,000 training iterations were used using a batch size of 32. In this setup, the knowledge is directly transfered between the first fully connected layers of the networks. The proposed "KT + Distillation" approach improves the accuracy over 2.5% when the NCC classifier is used. Again, the proposed knowledge transfer procedure improves the results over both the baseline networks and the distillation approach highlighting the effectiveness of the proposed knowledge transfer method.

V. Conclusions

In this paper, a method that is capable of transferring the knowledge between any two layers of neural networks was proposed. The proposed method is model-agnostic overcoming several limitations of existing knowledge transfer techniques, since the knowledge is transfered between layers that can have different architecture and no information about the donor

Transfer Set	Distillation		Knowledge Transfer	Knowledge Transfer + Distillation
	NCC	NN	NCC	NN
CIFAR-10 (30)	84.17%	80.28%	78.32%	77.56%	**75.57%**
CIFAR-10 (30) + $\mathcal{N}(0, 0.01)$	83.10%	79.55%	78.48%	78.35%	**74.59%**
CIFAR-10 (F)	74.64%	31.38%	69.59%	71.96%	**27.50%**

TABLE IV: CIFAR Evaluation: Knowledge transfer using different transfer sets (classification error rate)

Method	Accuracy (NCC)	Accuracy (NN)
Distillation	78.21%	81.76%
KT	77.99%	-
KT + Distill.	**80.81%**	**83.11%**

TABLE V: AFLW Evaluation: Knowledge transfer evaluation using different methods

model is required, except for the representation extracted from the corresponding layer. This allows the proposed method to overcome the limitations of other existing methods, e.g., requiring access to the weights of the donor model or keeping the same dimensionality between the layers used for knowledge transfer. The effectiveness of the proposed approach was demonstrated using three image datasets, including a large-scale dataset for learning a light-weight model for facial pose estimation that can be directly deployed on devices with limited computational resources, such as embedded systems for drones.

There are several interesting future research directions. First, the proposed method is also capable of providing hints for different layers of the donor network, as in [5], without the added complexity (and possible information loss) of dimensionality reduction. Also, augmenting the transfer set with noise seems to significantly improve the performance of the method. Therefore augmenting the transfer set with data from a similar domain and/or using synthetic data (e.g., learning the dataset that is optimal for knowledge transfer, similarly to [19]) is expected to further improve the knowledge transfer. Furthermore, transferring the knowledge from various layers of a large convolutional network to a smaller network might allow for learning very fast feature extractors that could be used for extracting representations than can be then fine-tuned for information retrieval [20] or clustering tasks [21]. Finally, the proposed methodology could be also applied to transfer the knowledge from a model of a biological system, where the training data are usually unknown, to a neural network just by using random noise or cross-domain data.

ACKNOWLEDGMENT

This project has received funding from the European Unions 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] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[2] C. Bucilu, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2006, pp. 535–541.

[3] T. Chen, I. Goodfellow, and J. Shlens, "Net2net: Accelerating learning via knowledge transfer," *arXiv preprint arXiv:1511.05641*, 2015.

[4] N. Passalis and A. Tefas, "Dimensionality reduction using similarity-induced embeddings," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2017.

[5] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[6] N. Passalis and A. Tefas, "Concept detection and face pose estimation using lightweight convolutional neural networks for steering drone video shooting," in *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 71–75.

[7] W. Chan, N. R. Ke, and I. Lane, "Transferring knowledge from a rnn to a dnn," *arXiv preprint arXiv:1504.01483*, 2015.

[8] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5900–5904.

[9] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[10] Z. Tang, D. Wang, Y. Pan, and Z. Zhang, "Knowledge transfer pre-training," *arXiv preprint arXiv:1506.02256*, 2015.

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

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

[13] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.

[14] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[15] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[16] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[18] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] D. Maclaurin, D. K. Duvenaud, and R. P. Adams, "Gradient-based hyperparameter optimization through reversible learning." in *Proceedings of the International Conference on Machine Learning*, 2015, pp. 2113–2122.

[20] 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.

[21] ——, "Information clustering using manifold-based optimization of the bag-of-features representation," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 52–63, 2018.