Probabilistic Knowledge Transfer for Lightweight Deep Representation Learning

Nikolaos Passalis, Maria Tzelepi and Anastasios Tefas

***Abstract*—Knowledge Transfer (KT) methods allow for transferring the knowledge contained in a large deep learning model into a more lightweight and faster model. However, the vast majority of existing KT approaches are designed to handle mainly classification and detection tasks. This limits their performance on other tasks, such as representation/metric learning. To overcome this limitation a novel Probabilistic KT (PKT) method is proposed in this paper. PKT is capable of transferring the knowledge into a smaller student model by keeping as much information as possible, as expressed through the teacher model. The ability of the proposed method to use different kernels for estimating the probability distribution of the teacher and student models, along with the different divergence metrics that can be used for transferring the knowledge, allows for easily adapting the proposed method to different applications. PKT outperforms several existing state-of-the-art KT techniques, while it is capable of providing new insight into KT by enabling several novel applications, as it is demonstrated through extensive experiments on several challenging datasets.**

***Index Terms*—Neural Network Distillation, Representation Learning, Metric Learning, Lightweight Deep Learning, Knowledge Transfer**

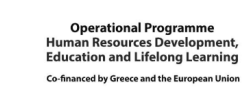
I. Introduction

Deep Learning (DL) allowed for tackling many difficult problems with great success [1], ranging from challenging computer vision problems [2], to complex reinforcement learning problems [3]. However, DL suffers from significant limitations, since neural networks are becoming increasingly complex, often composed of hundred of layers [4] and requiring powerful and dedicated hardware for effectively deploying them. This limits their applications on embedded and mobile devices with limited computational resources, e.g., memory, processing power, etc. These limitations shifted the research interest into developing more efficient models, which can effectively lower the computational and energy requirements of DL. To this end, several methods have been recently proposed, including model compression and quantization methods [5], which can reduce the number of bits needed to store the weights of a network and accelerate the required computations, as well as lightweight and more efficient neural network architectures [6].

Another promising line of research for further improving the performance of lightweight DL models are *Knowledge Transfer (KT)* methods [7], [8], which are also known as Knowledge Distillation (KD) or Neural Network Distillation methods. KT works by modeling the *knowledge*, as encoded by a large deep learning model, and then transferring it into a smaller and faster model. The most straightforward way to perform KT is to train the smaller model, which is called *student model*, to *mimic* the response (output) of a larger and more complex neural network, typically called *teacher model*. In this way, KT allows for training more accurate student models, that are able to generalize better compared with models trained without KT, since the teacher implicitly captures more information for each data sample and the training classes. Note that this information is usually not available when training with traditional methods, since only hard binary labels are provided, instead of a smooth probability distribution over the classes. This allows KT to better *regularize* the training process, significantly improving the performance of the student network [9]. It is worth noting that KT methods are orthogonal to other approaches used for developing lightweight models, e.g., they can be used with both quantization methods and lightweight architectures, such as [6]. This allows KT to be combined with any other method, further improving the performance of lightweight DL models.

Several KT methods have been proposed and successfully used for a wide variety of tasks, mainly related to classification [7], [8], [10] and object detection [11]. However, these approaches suffer from a series of limitations that prohibit them to be efficiently used for other DL-related tasks, such as representation/metric learning [12]. First, most of the existing methods are not capable of efficiently transferring the knowledge when a different number of neurons are used, i.e., the layers have different dimensionality. The main reason for this is that most KT methods are currently tailored towards classification tasks, where they are usually used to transfer the knowledge between the final classification layer of two networks, assuming that the dimensionality of these layer is the same for both networks. However, this renders most of the existing KT methods not suitable for representation learning tasks, such as text, image and multimedia information retrieval [13], [14], [15]. At the same time, there are many other applications requiring accurate, yet lightweight feature extractors, e.g., reducing the communication overhead between mobile devices and cloud [16], protecting the privacy of users by processing most of the data on the edge [17], [18], etc.

Nikolaos Passalis, Maria Tzelepi and Anastasios Tefas are with the Department of Informatics, Aristotle University of Thessaloniki, Greece. E-mail: passalis@csd.auth.gr, mtzelepi@csd.auth.gr, tefas@csd.auth.gr. This research is co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning 2014-2020" in the context of the project "Lighweight Deep Learning Models for Signal and Information Analysis" (MIS 5047925).



Furthermore, most of the existing KT methods usually completely ignore the actual *geometry* of the teacher's feature space, e.g., manifolds that are possibly formed, the local structure of the space (as expressed by the similarities between neighboring samples), etc., since they directly regress the representation extracted from the teacher model. However, it has been demonstrating that using this information allows for improving the quality of the learned model/representation for many different domains and applications [19], [20].

These observations lead us to a number of interesting questions: a) Can we use the existing KT methods for representation learning tasks and how do they perform on these tasks? b) Is it possible to design a method that can directly recreate the geometry of teacher's feature space using the student model? This would allow for effectively modeling the manifolds formed in the feature space of the teacher model and then recreating them into the student's lower dimensional feature space, possibly allowing for further improving the performance of the student. c) Can handcrafted features, e.g., HoG [21], be used to transfer the knowledge encoded in their representation into a neural network. This process can allow for effectively exploiting the enormous number of available unlabeled training data for training DL models. d) Finally, is it possible to model the gradual transformation of the feature space, through the various layers of the student model, in order to further improve the performance of the teacher model? If so, how we should match the layers between the student and teacher, e.g., should we directly transfer the knowledge between all the layers of the student and teacher or should we use a more adaptive and sophisticated approach to avoid over-regularizing the network?

In this paper a *Probabilistic* Knowledge Transfer (PKT) method is proposed, allowing for overcoming the limitations of existing methods and providing an efficient approach for developing lightweight DL models for various representation/metric learning tasks. To this end, the proposed method matches the probability distribution of the data in the feature spaces formed by the teacher and student models, instead of merely regressing their actual representation, as shown in Figure 1. PKT is motivated by the observation that matching the probability distributions in the feature space of the teacher and student models allows for maintaining the teacher's *Quadratic Mutual Information* (QMI) [22] in the smaller student model. Even though a set of labels is employed for modeling the MI of the model, no labels are actually required for the KT process, rendering the proposed approach fully unsupervised. This is possible, since, as it is thoroughly described in Section III, the aforementioned process leads to the unsupervised modeling of the interactions between the data samples in the feature space as a *probability distribution* that expresses the affinity between the data samples.

As we extensively demonstrate through several experiments, this process provides significant advantages over existing KT techniques. First, the proposed PKT method enables us to directly transfer the knowledge even when the output dimensionality of the networks does not match without using any additional dimensionality reduction layers. Furthermore, PKT allows for using the most appropriate kernel to model the

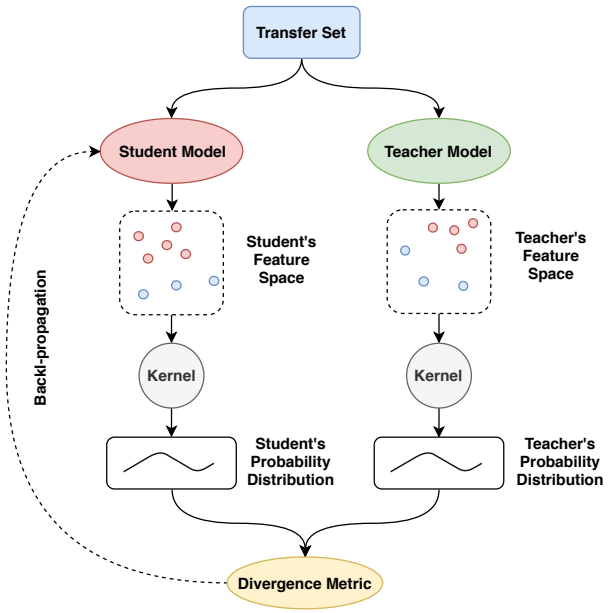


Fig. 1. Probabilistic Knowledge Transfer: First, the teacher's and student's knowledge is modeled using probability distributions. The knowledge is then transferred to the student by minimizing the divergence between these two probability distributions.

feature space of the teacher and student models, providing a straightforward way to finetune the proposed method for different applications, e.g., information retrieval using angular metrics, such as the cosine similarity. Finally, it is demonstrated that the probability distributions can also be estimated or enhanced using any other information source, such as handcrafted feature extractors, intermediate neural layers, supervised information or even qualitative information provided by users and/or domain experts. Thus, PKT constitutes a powerful and flexible KT tool that can be used to improve KT for existing scenarios, as well as support a number of novel KT scenarios. The proposed method can effectively a) perform cross-modal KT, b) transfer the knowledge encoded in handcrafted feature extractors into neural networks, c) transfer the knowledge into multiple layers using a hierarchical ladder scheme, d) learn representations robust to distribution shifts, and e) improve the performance for a wide variety of tasks (retrieval, classification, clustering). The proposed method is extensively evaluated and compared to other KT techniques using four challenging computer vision datasets and several evaluation setups (KT from deep neural networks, KT from handcrafted feature extractors, KT from different modalities (cross modal transfer), hierarchical KT from multiple layers).

This paper is an extended version of our previous work presented in [23]. We further extended our previous paper by a) providing a multi-kernel formulation that improves the performance of the proposed method over different setups, b) employing a *ladder*-based KT scheme that allows for effectively transferring the knowledge between multiple levels of two networks, despite the difficulties that often arise from this process, c) providing an extensive stability and ablation study, d) providing additional comparison with a recently

proposed metric learning approach [24] and e) evaluating the proposed method in additional setups (classification, distribution shift and unsupervised knowledge discovery), as well as additional datasets (STL-10 [25] and ILSVRC [26]). An open-source implementation of the proposed method, containing all the improvements proposed, in this paper will be publicly available at https://github.com/passalis/deep_pkt.

The rest of the paper is structured as follows. First, in Section II the related works are discussed, while the proposed method is analytically presented in detail in Section III. Then, the proposed method is evaluated in Section IV. Finally, Section V concludes this paper.

II. Related Work

Research on knowledge transfer methods, which allow for effectively training smaller and faster models, is mainly driven by the increasing size and complexity of DL models, as well as the need to deploy them intro various devices with limited computing capabilities, such as embedded and mobile devices. A large portion of the KT methods proposed in the literature employ the teacher model to generate soft-labels and then using these soft-labels for training the smaller student network [8], [9], [27], [28], [29]. Indeed, one of the earliest approaches to KT directly used the label distribution predicted by the teacher to train the student [27], while the well-known neural network distillation method [8], extended this approach by appropriately tuning the temperature of the softmax activation. Neural network distillation can indeed effectively regularize the smaller network, leading to better performance than directly training the network using hard binary labels [6], [8], [15]. Several extensions to this approach have been proposed: soft-labels can be used for pre-training a large network [30], used for domain adaptation tasks [28], for *compressing* the posterior density in Bayesian methods [31], or for transferring the knowledge from recurrent neural networks into simpler models [29]. The regularization nature of neural network distillation is also highlighted in [9], where the knowledge was transferred from a smaller teacher model to a larger student network, allowing for training the student with fewer annotated data. These methods are mainly designed to handle classification tasks and cannot be effectively applied for representation learning, as it is also demonstrated in Section IV. It is also worth noting that KT is closely related to optimization methods [32], [33], [34], [35], since it performs a relaxation of the original optimization problem, as well as to various loss functions proposed for metric learning, such as [36].

A quite different approach was proposed in [7], where the student network was trained by employing *hints* from the intermediate layers of the teacher model. To this end, projection matrices were employed to match the dimensionality between the teacher's and student's layers, since usually the student model is smaller. Even though this approach allows for transferring the knowledge between arbitrary layers, it can lead to a significant loss of information, due to using low-dimensional projections. A similar approach was also employed in [38], where instead of using hints, the *flow of solution procedure* (FSP) matrix was used to transfer the knowledge between the intermediate layers of various residual networks. It is worth noting that FSP, in contrast with hints, cannot be applied when the intermediate representations have different size and, as a result, cannot be used for representation learning. Finally, in [20] an embedding-based approach was proposed for transferring the knowledge, while in [24], a multidimensional-scaling based method was used to learn a student that maintains the same distances as the teacher between pairs of samples. However, the first method is tailored toward classification tasks, while the latter leads to significantly worse performance compared to the proposed one (demonstrated in Section IV), since directly matching the distances in high-dimensional spaces is less effective than the proposed probabilistic formulation which can alleviate these limitations (especially when coupled with a carefully selected kernel function and divergence metric).

To the best of our knowledge, the proposed method is the first probabilistic KT approach that can be effectively used for a wide range of different representation learning tasks. Indeed, the proposed method can be employed in many different and novel scenarios, e.g., transferring the knowledge from handcrafted feature extractors, as it is demonstrated in Section IV. The proposed method does not require extensive hyper-parameter turning, such as the tedious tuning of the softmax temperature [8], while it is easy to implement and use. At the same time, its ability to estimate the probability distribution using different kernels, as well as to combine them, allows for learning more robust lightweight student models. Furthermore, the proposed method can directly handle layers of different dimensionality, without requiring using additional dimensionality reduction layers [7] or less efficient distance-based matching [24]. Finally, also note that a KT approach that employs an intermediate network was also proposed in [37]. However, compared to this approach, the proposed one is designed to facilitate multi-layer KT transfer.

III. Probabilistic Knowledge Transfer

First, the used notation and required background are briefly introduced in this Section. Then, the proposed method is analytically described. Several design choices are discussed through this Section, along with extensions that allow for handling several different KT scenarios.

A. Notation and Background

Let $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N\}$ be a *transfer* set of N objects. The transfer set is used to transfer the knowledge encoded in the teacher model into the student model. The notation $\mathbf{x} = f(\mathbf{t})$ and $\mathbf{y} = g(\mathbf{t}, \mathbf{W})$ is used to refer to the teacher's and student's output representation (respectively). Note that $\mathbf{W}$ refers to the trainable parameters of the student model. The student model $g(\cdot)$ is then trained in order to "mimic" the behavior of $f(\cdot)$. Note that there is no constraint on what the functions $f(\cdot)$ and $g(\cdot)$ are as long as the output of $f(\cdot)$ is known for every element of $\mathcal{T}$ and $g(\cdot)$ is a differentiable function. Actually, the first constraint can be relaxed even more, since, as it will be demonstrated later, it

is enough to know just the pairwise similarities between the training samples for the teacher model. The continuous random variables X and Y are used to model the distribution of the representation extracted from the teacher and student models. Also, an additional discrete random variable C is introduced to describe some higher level semantic features of the samples, e.g., class labels. Therefore, each vector $\mathbf{x}$ drawn from X is also associated with an attribute label c.

Mutual Information (MI) is a measure of uncertainty regarding the class label c after observing the corresponding vector $\mathbf{x}$ [22]. Let $P(c)$ be the probability of observing the class label c. The teacher's MI can be formally defined as:

$$I(X,C) = \sum_{c} \int_{\mathbf{x}} p(\mathbf{x},c) \log \frac{p(\mathbf{x},c)}{p(\mathbf{x})P(c)} d\mathbf{x}, \tag{1}$$

where $p(\mathbf{x},c)$ is the joint probability density function of $\mathbf{x}$ and c. Then, Quadratic Mutual Information (QMI) can be derived by replacing the KL divergence between $p(\mathbf{x},c)$ and the product of marginal probabilities that appear in (1) by the quadratic divergence measure [22]:

$$I_T(X,C) = \sum_{c} \int_{\mathbf{x}} (p(\mathbf{x},c) - p(\mathbf{x})P(c))^2 d\mathbf{X}. \tag{2}$$

By expanding (2), QMI can be more compactly expressed in terms of three quantities, called *information potentials*, as: $I_T(X,C) = V_{IN} + V_{ALL} - 2V_{BTW}$, where the corresponding potentials are defined as: $V_{IN} = \sum_c \int_{\mathbf{x}} p(\mathbf{x},c)^2 d\mathbf{x}$, $V_{ALL} = \sum_c \int_{\mathbf{x}} (p(\mathbf{x})P(c))^2 d\mathbf{x}$, and $V_{BTW} = \sum_c \int_{\mathbf{x}} p(\mathbf{x},c)p(\mathbf{x})P(c) d\mathbf{x}$. The potential V_{IN} expresses the in-class interactions, the potential V_{ALL} the interactions between all the samples, while the potential V_{BTW} the interaction of each class against all the other samples, as further shown in (5)-(7).

The class prior probability for each c_p class can be estimated as $P(c_p) = \frac{J_p}{N}$, where J_p refers to the number of samples for the p-th class and N is the size of transfer set. Then, Kernel Density Estimation (KDE) can be employed for estimating the joint density probability as:

$$p(\mathbf{x},c_p) = p(\mathbf{x}|c_p)P(c_p) = \frac{1}{N}\sum_{j=1}^{J_p} K(\mathbf{x},\mathbf{x}_{pj},\sigma^2), \tag{3}$$

where $K(\mathbf{a},\mathbf{b},\sigma^2)$ is a symmetric kernel with width σ and the notation $\mathbf{x}_{pj}$ is used to refer to the j-th sample of the p-th class. The density of X is similarly estimated as:

$$p(\mathbf{x}) = \sum_{p=1}^{J_p} p(\mathbf{x},c_p) = \frac{1}{N}\sum_{j=1}^{N} K(\mathbf{x},\mathbf{x}_j,\sigma^2). \tag{4}$$

B. Probabilistic Knowledge Transfer

The knowledge can be transferred from the teacher model to the student model by maintaining the same amount of MI between the two random variables X and Y and the class labels, i.e., $I(X,C) = I(Y,C)$. In the case of QMI, this implies that the information potentials of the student model should be equal to the corresponding information potentials of the teacher. Note that there might be other configurations with equal MI as well, however it is enough to use one of them to transfer the knowledge.

The information potentials for the teacher model can be easily calculated using the probabilities estimated in the previous subsection as:

$$V_{IN}^{(t)} = \frac{1}{N^2}\sum_{p=1}^{N_c}\sum_{k=1}^{J_p}\sum_{l=1}^{J_p} K(\mathbf{x}_{pk},\mathbf{x}_{pl},2\sigma_t^2), \tag{5}$$

$$V_{ALL}^{(t)} = \frac{1}{N^2}\left(\sum_{p=1}^{N_c}(\frac{J_p}{N})^2\right)\sum_{k=1}^{N}\sum_{l=1}^{N} K(\mathbf{x}_k,\mathbf{x}_l,2\sigma_t^2), \tag{6}$$

and

$$V_{BTW}^{(t)} = \frac{1}{N^2}\sum_{p=1}^{N_c}\frac{J_p}{N}\sum_{j=1}^{J_p}\sum_{k=1}^{N} K(\mathbf{x}_{pj},\mathbf{x}_k,2\sigma_t^2), \tag{7}$$

where N_C is the total number of classes. The interaction between two samples $\mathbf{t}_i$ and $\mathbf{t}_j$ is measured using the kernel function $K(\mathbf{x}_i,\mathbf{x}_j,\sigma^2)$ that expresses the similarity between them (as measured through the representation extracted from the teacher model). Similarly, the information potentials for the student network are defined: $V_{IN}^{(s)} = \frac{1}{N^2}\sum_{p=1}^{N_c}\sum_{k=1}^{J_p}\sum_{l=1}^{J_p} K(\mathbf{y}_{pk},\mathbf{y}_{pl},2\sigma_s^2)$, $V_{ALL}^{(s)} = \frac{1}{N^2}(\sum_{p=1}^{N_c}(\frac{J_p}{N})^2)\sum_{k=1}^{N}\sum_{l=1}^{N} K(\mathbf{y}_k,\mathbf{y}_l,2\sigma_s^2)$, and $V_{BTW}^{(s)} = \frac{1}{N^2}\sum_{p=1}^{N_c}\frac{J_p}{N}\sum_{j=1}^{J_p}\sum_{k=1}^{N} K(\mathbf{y}_{pj},\mathbf{y}_k,2\sigma_s^2)$. Different (and appropriately tuned) bandwidths σ_t and σ_s must be used for the teacher and student models, since the kernels are used to model the distribution in two different feature spaces. The bandwidth of the student model was set to 1, while the bandwidth of the teacher was determined according to the mean distance between the samples in the teacher's feature space. This strategy was used for all the conducted experiments, follows the experimental findings of other related approaches [39], and ensures that a meaningful probability estimation, i.e., that the kernel values will not collapse to either 0 (too small bandwidth) or 1 (too large bandwidth), will be obtained for both models.

It is easy to see that the most straightforward way to ensure that the information potentials will be equal among the two models is to require the similarity between each pair of points, as expressed through the employed kernel, to be equal:

$$K(\mathbf{x}_i - \mathbf{x}_j, 2\sigma_t^2) = K(\mathbf{y}_i - \mathbf{y}_j, 2\sigma_s^2)\ \forall i,j. \tag{8}$$

Therefore, the problem of transferring the knowledge by maintaining the same MI between the models and a set of classes is reduced into matching the kernel values between different pairs of data. PKT, instead of matching the unnormalized kernel, proposes to minimize the divergence between the teacher's and student's conditional probability distributions:

$$p_{i|j}^{(t)} = \frac{K(\mathbf{x}_i,\mathbf{x}_j,2\sigma_t^2)}{\sum_{i=1,i\neq j}^{N} K(\mathbf{x}_i,\mathbf{x}_j,2\sigma_t^2)} \in [0,1], \tag{9}$$

and

$$p_{i|j}^{(s)} = \frac{K(\mathbf{y}_i,\mathbf{y}_j,2\sigma_t^2)}{\sum_{i=1,i\neq j}^{N} K(\mathbf{y}_i,\mathbf{y}_j,2\sigma_s^2)} \in [0,1]. \tag{10}$$

TABLE I
DIFFERENT KERNELS THAT CAN BE USED FOR TRANSFERRING THE KNOWLEDGE BETWEEN THE TEACHER AND STUDENT MODELS

Kernel	Notation	Parameters	Expression
Gaussian	K_g	σ	$exp(-\frac{\|\|\mathbf{a}-\mathbf{b}\|\|_2^2}{\sigma^2})$
Cosine	K_c	-	$\frac{1}{2}(\frac{\mathbf{a}^T\mathbf{b}}{\|\|\mathbf{a}\|\|_2\|\|\mathbf{b}\|\|_2}+1)$
T-student	K_s	d	$\frac{1}{1+\|\|\mathbf{a}-\mathbf{b}\|\|_2^d}$

These probabilities express how probable is for each sample to select each of its neighbors [39], modeling in this way the geometry of the feature space.

Several different kernel functions can be used to estimate the corresponding probabilities, while some of possible choices are listed in Table I, where the kernels are applied on two vectors $\mathbf{a}$ and $\mathbf{b}$ and the notation $||\cdot||_2$ is used to refer to the l_2 norm of a vector. Among the most popular ones is the Gaussian kernel [40]. However, Gaussian kernels require to carefully tune their width, which is not a straightforward task. It is worth noting that several heuristics have been proposed to address specifically this limitation [41]. In the initial version of PKT [23], this issue was avoided by deriving an angular kernel that requires no domain-dependent tuning (cosine kernel). This kernel also fits various retrieval tasks especially well, since the cosine similarity is usually used for retrieval. However, as we experimentally found out, this kernel is not optimal for every task. Instead, l^2-based kernels, such as the Gaussian and T-student kernels, seem to perform better on such tasks, e.g., clustering and classification. This behavior is experimentally demonstrated in the results reported in Section IV.

Therefore, in this work we propose using a hybrid objective that requires minimizing the divergence calculated using both the cosine kernel, which ensures the good performance of the learned representation for retrieval tasks, and the T-student kernel, which ensures the good performance of the method for classification tasks:

$$\mathcal{L} = \mathcal{D}(\mathcal{P}_c^{(t)}, \mathcal{P}_c^{(s)}) + \mathcal{D}(\mathcal{P}_T^{(t)}, \mathcal{P}_T^{(s)}), \tag{11}$$

where $D(\cdot)$ is a divergence metric and the notation $\mathcal{P}_c^{(t)}$ and $\mathcal{P}_T^{(t)}$ is used to denote the conditional probabilities of the teacher calculated using the cosine and T-student kernels respectively. The student probability distribution is denoted similarly by $\mathcal{P}_c^{(s)}$ and $\mathcal{P}_T^{(s)}$. It is worth noting that the proposed method only requires one additional feed-forward pass for the teacher model in order to extract the teacher's representation and calculate the loss $\mathcal{L}$. The complexity of calculating the loss is $O(N_B^2 N_d)$ where N_B is the batch size and N_d is the size of the representation extracted from the teacher/student (or the maximum dimensionality among them, if they are different).

There are also several different choices for defining the divergence metric. In this work, a symmetric version of the Kullback-Leibler (KL) divergence, the Jeffreys divergence [42], is used to this end:

$$\begin{aligned}\mathcal{D}_J(\mathcal{P}^{(t)}||\mathcal{P}^{(s)}) = \int_{-\infty}^{+\infty} (\mathcal{P}^{(t)}(\mathbf{t}) - \mathcal{P}^{(s)}(\mathbf{t})) \\ \left(\log \mathcal{P}^{(t)}(\mathbf{t}) - \log \mathcal{P}^{(s)}(\mathbf{t})\right) d\mathbf{t},\end{aligned} \tag{12}$$

The final divergence function, that can be readily used for training the model, is defined as:

$$\begin{aligned}\mathcal{D}_J(\mathcal{P}^{(t)}||\mathcal{P}^{(s)}) = \\ \sum_{i=1}^{N}\sum_{j=1,i\neq j}^{N} \left(p_{j|i}^{(t)} - p_{j|i}^{(s)}\right) \cdot \left(\log p_{j|i}^{(t)} - \log p_{j|i}^{(s)}\right),\end{aligned} \tag{13}$$

since the two distribution are sampled using a finite number of points. Note that other metrics, such as the KL divergence, can be employed to meet the requirements of each application. For example, KL is an asymmetric metric which can be used when higher weight should be given to minimize the divergence for neighboring pairs of points instead of distant ones.

To learn the parameters $\mathbf{W}$ of the student model $g(\mathbf{t}, \mathbf{W})$ gradient descent is used: $\Delta\mathbf{W} = -\eta\frac{\partial\mathcal{L}}{\partial\mathbf{W}}$, where $\mathbf{W}$ is the matrix with the parameters of the student model and η is the employed learning rate. The derivative of the loss function with respect to the parameters of the model can be easily derived by observing that:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}} = \sum_{i=1}^{N}\sum_{j=1,i\neq j}^{N} \frac{\partial\mathcal{L}}{\partial p_{j|i}^{(s)}} \sum_{l=1}^{N} \frac{\partial p_{j|i}^{(s)}}{\partial \mathbf{y}_l} \frac{\mathbf{y}_l}{\partial\mathbf{W}}, \tag{14}$$

where $\frac{\mathbf{y}_l}{\partial\mathbf{W}}$ is just the derivative of the student's output with respect to its parameters. Furthermore, the conditional probabilities can be estimated using relatively small batches of the data at each iteration, e.g., batches of 128 samples, instead of using the whole dataset. In this way, it is possible to accelerate the convergence of the method, while it was experimentally established that this batch-based approach does not negatively impact the learned representation. Note that to ensure that different sample pairs are used for each iteration, the transfer samples are shuffled after each epoch. Finally, note that the teacher model can be any model for which we can estimate the conditional probability $p_{j|i}^{(t)}$. Therefore, the source of knowledge for the teacher can range from representations extracted from neural layers and handcrafted features (where kernels are used to estimate the probabilities) to domain knowledge and label attributes. In the latter case, it is worth noting that only the pairwise similarities between the samples are required (allowing for transferring the knowledge even when there is no representation extracted from the teacher for each data sample).

C. Ladder Probabilistic Knowledge Transfer

The proposed method can be directly used to transfer the knowledge from multiple layers of two neural networks. This is expected to better guide the knowledge transfer process, similarly to the way that hints from multiple layers guide the neural network distillation process [7]. However, if the layers from and to which the knowledge will be transferred are not carefully selected the student network can be over-regularized leading to worse performance compared to not using intermediate layers for the KT, as we also experimentally demonstrated in the ablation study provided in Section IV. Note that selecting the appropriate layers can be a quite difficult and tedious process that involves several experiments

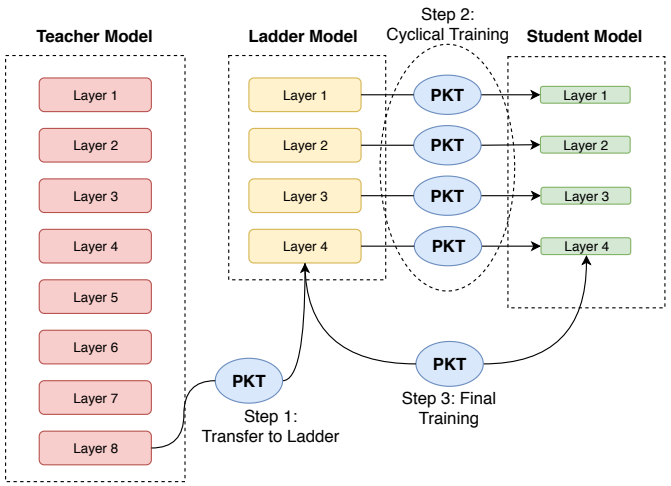


Fig. 2. Ladder PKT: Constructing an intermediate *ladder* network allows for more effectively transferring the knowledge to all the layers of the student model. First, the knowledge is transferred to the intermediate ladder network. Then, PKT is employed in a two step process: a) the knowledge is transferred between all the intermediate layers and b) the final representation (extracted from the last layer of the networks) is further fine-tuned.

to find the best combination of layers, especially when the architectures of the student and teacher differ a lot.

To overcome this limitation, we propose a simple, yet effective ladder-based approach for transferring the knowledge between all the layers of the student and an appropriately constructed model, as shown in Fig. 2. First, an intermediate network, called *ladder network*, with the same number of layers as the student model is employed. This allows for having a one-to-one matching between each layer of the student and ladder models, overcoming the need to carefully select and match the layers between the teacher and ladder models. At the same time, the ladder network should contain more parameters, i.e., the number of neurons/convolutional filters per layer must be increased, to ensure that enough knowledge will be always available to the ladder network (when compared to the student model).

First, the knowledge is transferred from the final layer of the teacher model to the final layer of the ladder network, by performing regular PKT. Then, the knowledge can be readily transferred from the ladder to the student using all the available layers, since, the ladder and student networks have similar architecture, and, as a result, all the layers can be used without the risk of mismatching between the layers. Finally, instead of directly transferring the knowledge from all the layers at once, we propose using a *cyclical* schedule, i.e., select a random layer pair (of the same depth) at each training epoch and transfer the knowledge between the layers of this pair. This process provides greater flexibility and allows for better exploring the solution space. Then, the KT process continues by employing regular PKT to transfer the knowledge to the final representation layer.The complexity of calculating the loss in the case of multi-layer KT is updated as $O(KN_B^2N_D)$, where K is the number of layers used for the KT and N_D refers to the maximum dimensionality of any intermediate layer.

TABLE II
COMPUTATIONAL COMPLEXITY OF THE MODELS USED FOR THE CONDUCTED EXPERIMENTS

Model	# Parameters	FLOPS
CIFAR - Teacher	6.95k	0.17M
CIFAR - Ladder	57.99k	0.78M
CIFAR - Teacher	11.17M	278.32M
ILSCVR - Teacher	2.03M	69.02M
ILSCVR - Student	0.23M	5.90M
SUN Attribute - Student	77.73k	6.02M

(k refers to 10^3, M refers to 10^6)

TABLE III
CIFAR-10: RETRIEVAL EVALUATION

Method	mAP (e)	mAP (c)	t-50 (e)	t-50 (c)
Teacher	87.18	90.47	92.40	92.45
Student	41.41	47.36	68.86	72.50
Training from scratch (70+30 epochs / $\eta = 10^{-3}/10^{-4}$)				
Hint (random proj.) [7]	58.06	65.27	73.89	75.43
Hint (optimized proj.) [7]	54.62	61.15	74.57	76.20
Distillation [8]	40.94	46.52	68.87	72.51
MDS-T [24]	51.86	54.63	68.06	70.13
PKT	**62.45**	**66.83**	**75.10**	**77.09**
Pre-trained student (30+10 epochs / $\eta = 10^{-3}/10^{-4}$)				
Hint (random proj.) [7]	57.00	64.33	72.93	74.66
Hint (optimized proj.) [7]	53.81	61.09	74.47	76.91
Distillation [8]	40.59	46.54	68.95	72.71
MDS-T [24]	54.49	57.40	71.28	73.37
PKT	**62.12**	**66.78**	**75.02**	**76.97**

IV. EXPERIMENTAL EVALUATION

The proposed method is extensively evaluated and compared to other baseline and state-of-the-art knowledge transfer methods in this Section. First, the proposed method is evaluated under a representation learning setup using the CIFAR-10 dataset [43], as well as a challenging subset of the large-scale ILSVRC dataset [26]. Then, the proposed method is evaluated under a distribution shift setup, using the STL-10 dataset [25], as well as by transferring the knowledge encoded in handcrafted feature extractors and different modalities using the SUN Attribute dataset [44]. Finally, a sensitivity analysis is provided, where the effect of the employed kernel on the performance on the learned representations is evaluated. Please note that details regarding the experimental setup, such as the employed network architectures, optimization hyper-parameters, etc., are provided in the supplementary material. The computational complexity (time and space complexity) of the networks employed for the conducted experiments is summarized in Table II. Note that a wide variety of different networks and transfer setups is employed. No information regarding the teacher model is provided for the SUN Attribute dataset, since handcrafted features are used for the knowledge transfer for the experiments conducted using this dataset.

Representation learning using CIFAR-10: The experimental results using the CIFAR-10 dataset are reported in Table III. A content-based image retrieval setup was employed for the evaluation. The database contains the training image, while the test set images are used for evaluating the performance

TABLE IV
CIFAR-10: CLUSTERING EVALUATION

Method	Rand	MI	V-score	FM	CH
Teacher	0.831	0.832	0.832	0.848	3882.9
Student	0.427	0.527	0.532	0.487	659.2
Hint (random) [7]	0.483	0.587	0.599	0.542	2297.5
Hint (optim.) [7]	0.400	0.561	0.577	0.473	1603.2
Distillation [8]	0.407	0.513	0.518	0.469	637.5
MDS-T [24]	0.538	0.589	0.592	0.585	1072.8
PKT	**0.580**	**0.626**	**0.630**	**0.623**	**2323.5**

TABLE V
CIFAR-10: LADDER TRANSFER EVALUATION

Method	mAP (e)	mAP (c)	t-50 (e)	t-50 (c)
Teacher	87.18	90.47	92.40	92.45
Ladder Teacher	62.12	66.78	75.02	76.97
Student	29.15	31.79	47.75	49.93
Transfer from Teacher (ResNet-18) 80 + 20 epochs / $\eta = 10^{-3}/10^{-4}$				
Hint (random) [7]	34.11	37.61	47.01	48.36
Hint (optim.) [7]	32.28	36.71	50.41	52.67
Distillation [8]	28.44	31.26	48.17	50.47
MDS-T [24]	30.36	31.99	45.12	46.76
PKT	37.05	40.09	50.28	52.96
Transfer from Ladder Teacher (PKT from ResNet-18) 50 cyclical transfer epochs / $\eta = 10^{-3}$				
Hint (optim.) [7]	32.94	35.80	50.39	51.95
MDS-T [24]	34.35	36.59	48.39	50.52
PKT	**37.72**	**40.59**	**50.69**	**53.27**

of each model. The (interpolated) mean Average Precision (mAP) at the standard 11-recall points and the top-k precision (abbreviated as "t-k") were used as the evaluation metrics [13]. Both the Euclidean distance (denoted by "e") and the cosine similarity (denoted by "c") were used for the retrieval process. Also, note that cosine similarity is actually equivalent to the Euclidean distance, when used to rank the similarity between a query and the documents in the database, when l_2 normalization is performed on the extracted feature vectors. In this way, these two metrics provide a way to evaluate the performance of the learned representation with and without using normalization.

The penultimate layers were used to extract a representation from each image for both the teacher (512-dimensional features) and student (128-dimensional features) models. The proposed method was compared to two variants of the hint-based knowledge transfer method [7], abbreviated as "Hint" (in the first one a random projection was used, while in the second one the projection was optimized along with the student model). Furthermore, the multidimensional scaling-based method proposed in [24] was also evaluated (abbreviated as "MDS-T"), along with the plain distillation approach (where the knowledge was transferred from the final classification layer). For the rest of the evaluated methods, the knowledge was transferred from the penultimate layer of the teacher network (512 dimensions) to the corresponding layer of the student network (128 dimensions).

Several conclusions can be drawn from the results reported in Table III. First, the proposed method leads to significant improvements compared to both the baseline student network trained for classification (the mAP increases from 47.36% to 66.83%), as well as to all the other evaluated methods regardless the metric used for the retrieval process. Also, training from scratch seems to lead to slightly better solutions, compared to continuing the training process using the pre-trained student network for most methods (including the proposed one). Quite interestingly, using random projections for the hint-based method leads to consistently better results, compared to optimizing the projection. This is perhaps due to the fact that the projection is becoming a vital part of the network, and as a result, the learned representation losses part of its discrimination ability when the projection is removed.

The quality of the learned representations was also evaluated using a knowledge discovery setup, where the representations were clustered into 10 clusters using the k-means algorithms. Various clustering metrics were measured [45]: the adjusted Rand score (Rand), the adjusted mutual information (MI), the V-score measure (V-score), the Fowlkes-Mallows (FM) score, and the Calinski Harabasz (CH) score. The experiments were conducted using the representation learned using the pre-trained student (last block of Table III). The experimental results are reported in Table IV. It is worth noting that even though the MDS-T method did not perform well on the retrieval evaluation, it performed significantly better on the clustering evaluation. Nonetheless, the proposed method vastly outperformed all the other evaluated methods for all the evaluated clustering criteria.

Finally, the proposed ladder-based transfer is evaluated on Table V. To evaluate the quality of the ladder transfer we employed a student trained with the PKT method as the ladder network, while a new student network was created by removing half of the convolutional filters/neurons for each layer of the ladder network. Note that the proposed ladder transfer method can also be applied for the Hint and MDS methods by transferring the knowledge between all the layers of the ladder and teacher networks. The proposed ladder approach indeed improves the performance for all the methods that can be used to support ladder-based KT (hints with random projections did not convergence, so the corresponding results were not included). Note that significant improvements were obtained when the proposed ladder transfer method was combined with the MDS approach, for which the mAP increases from 31.99% to 36.59%. This highlights the importance of using the appropriate teacher models for training the students, regardless the employed KT method. Again, the proposed method outperforms all the other evaluated methods, leading to the higher mAP and top-50 precision.

Representation learning using ILSVRC: Next, the proposed method was evaluated under a more challenging setup, where the knowledge was transferred for a specific subset of the Imagenet containing 5 classes related to recognizing household appliances ("microwave", "dish washer", "refrigerator", "washer", and "vacuum"). The teacher model was trained from scratch, instead of using a larger network pre-trained on the whole Imagenet dataset, since we found out that all the methods performed significantly better when the knowledge was transferred from a network specialized on the specific

TABLE VI
IMAGENET SUBSET: RETRIEVAL EVALUATION

Method	mAP (e)	mAP (c)	t-50 (e)	t-50 (c)
Teacher	48.75	48.75	62.23	62.23
Student	39.50	39.50	53.63	53.63
Hint (random proj.) [7]	37.50	39.92	51.96	53.06
Hint (optimized proj.) [7]	34.27	36.24	50.25	52.17
Distillation [8]	36.85	40.55	53.70	54.84
MDS-T [24]	38.61	41.08	52.48	53.89
PKT	**41.04**	**44.71**	**55.94**	**57.34**

TABLE VII
CIFAR-10: RETRIEVAL EVALUATION USING THE STL-10 DATASET

Method	mAP (e)	mAP (c)	t-50 (e)	t-50 (c)
Teacher	57.40	61.20	68.87	71.36
Student	33.03	36.95	47.66	51.59
Transfer Set: CIFAR-10 (30+10 epochs / $\eta = 10^{-3}/10^{-4}$)				
Hint (random proj.) [7]	41.62	46.24	53.41	55.78
Hint (optimized proj.) [7]	39.84	44.88	53.32	56.76
Distillation [8]	32.60	36.54	47.37	51.52
MDS-T [24]	41.41	43.26	52.94	54.93
PKT	**44.89**	**48.48**	**56.13**	**58.48**
Transfer Set: STL (20+10 epochs / $\eta = 10^{-3}/10^{-4}$)				
Hint (random proj.) [7]	43.18	48.22	56.77	59.59
Hint (optimized proj.) [7]	41.88	47.21	56.69	60.06
Distillation [8]	36.51	40.89	52.46	56.52
MDS-T [24]	42.38	46.21	54.94	58.23
PKT	**45.61**	**49.63**	**57.57**	**60.77**

subset of classes. Despite the significantly different setup, e.g., different receptive fields for the convolutional layers, different number of classes, etc., the results are similar with those reported for the CIFAR-10 dataset. The proposed method outperforms all the other evaluated methods under all the evaluated metrics.

Distribution shift evaluation using CIFAR-10 and STL-10: The models trained using the CIFAR-10 dataset (KT using the pre-trained model) were also evaluated using a more challenging setup, where the STL-10 dataset, which contains the same classes as the CIFAR-10 (except for one), was employed to evaluate the KT methods in a distribution shift scenario. STL-10 images were resized to 32×32 pixels, in order to be compatible with the networks trained on CIFAR-10. The results are reported in Table VII. Two different scenarios were used: in the first one the transfer set was the CIFAR-10 dataset, while in the second one the transfer set was the STL-10 dataset. In both cases no labels were used for the KT process, while the teacher ResNet-18 model was trained using the CIFAR-10 dataset as before. The proposed method again outperforms all the other evaluated methods for both scenarios. As expected, using data from the same distribution for the KT process, i.e., using the training set of STL-10 dataset as the transfer set, leads to better retrieval precision for all the evaluated methods. Note again that no labels were needed for this process, allowing for using the whole unlabeled training set of the STL-10 dataset (100,000 images).

Cross-modal KT: The proposed method was also evaluated on a cross-modal KT setup using the SUN Attribute dataset [44] that contains more than 700 categories of scenes and 14,000 images, where each image is described by 102 discriminative textual attributes. The evaluation results are reported in Table VIII. First, the knowledge was transferred from 2×2 HoG features [44] into the student network. The proposed PKT method leads to 31.21% mAP outperforming all the other methods and achieving almost the same performance as the original HoG features. Furthermore, the proposed method was also evaluated under a cross-modal KT setup where the knowledge was transferred from the textual modality (expressed in the form of a list of textual attributes) into the student neural network that operates within the visual modality ("Attribute" features). Transferring the knowledge from the textual modality indeed improves the precision of the student network for all the evaluated methods, while the proposed PKT approach again outperforms the rest of the evaluated methods, demonstrating the flexibility of the proposed approach and its ability to support novel KT scenarios.

TABLE VIII
SUN ATTRIBUTE: CROSS-MODAL KNOWLEDGE TRANSFER

Method	Features	mAP (c)	top-50 (c)
HoG	-	32.06 ± 1.20	34.13 ± 1.64
Attribute	-	65.30 ± 1.99	67.15 ± 3.01
Hint (random proj.) [7]	HoG	22.83 ± 1.14	24.38 ± 1.72
Hint (optim. proj.) [7]	HoG	26.65 ± 3.82	27.90 ± 4.52
MDS-T [24]	HoG	29.62 ± 3.31	32.52 ± 4.56
PKT	HoG	$\mathbf{31.21 \pm 2.83}$	$\mathbf{33.38 \pm 3.11}$
Hint (random proj.) [7]	Attribute	39.55 ± 2.59	41.83 ± 3.05
Hint (optim. proj.) [7]	Attribute	43.38 ± 2.59	45.64 ± 3.29
MDS-T [24]	Attribute	36.85 ± 2.29	38.44 ± 2.63
PKT	Attribute	$\mathbf{47.22 \pm 5.20}$	$\mathbf{49.05 \pm 6.13}$

Sensitivity Analysis: Finally, we evaluated the effect of the kernel used for estimating the probability distributions to the effectiveness of the KT process. The results are reported in Table IX, using the same setup as in Table III (30 training epochs were used). The "Combined" method refers to the proposed combined loss that employs both the cosine and T-student kernels, as described in Section III. The Gaussian kernel is almost always the worst performing kernel. The T-student kernel performs well on the classification tasks (1-nearest neighbor classification accuracy - "1-NN"), as well as on retrieval tasks using the Euclidean distance as the affinity metric. On the other hand, the cosine kernel leads to significantly better results for retrieval using the cosine similarity, but to significantly worse when used for retrieval using the Euclidean distance. These results motivated our choice for combining both the cosine and T-student losses. Indeed, the proposed combined approach significantly improved the worst case performance, almost matching the best results for each individual kernel, while outperforming both of them on the 1-NN classification accuracy. It is worth noting that even when training with the Gaussian/T-student kernel, the best retrieval precision is acquired when the cosine similarity is used instead of the Euclidean distance. This somewhat unexpected finding demonstrates that the l_2 normalization, which is involved in the cosine metric, can lead to improved retrieval precision.

TABLE IX
EFFECT OF THE KERNEL FUNCTION ON PKT

Kernel	Pre-trained	1-NN (e)	mAP (e)	mAP (c)
Gaussian	Yes	78.75	50.51	49.14
Cosine	Yes	84.76	56.11	**66.52**
T-student	Yes	85.36	**61.66**	64.76
Combined	Yes	**85.46**	61.18	65.88
Gaussian	No	83.89	59.59	61.98
Cosine	No	84.02	54.01	**64.74**
T-student	No	84.83	**60.14**	63.00
Combined	No	**84.88**	59.80	64.04

TABLE X
LADDER ABLATION STUDY

Layers	Teacher	Cyclical Train.	mAP (e)	mAP (c)
4	ResNet	-	37.05	40.09
3-4	Ladder	Yes	37.28	40.21
2-3-4	Ladder	Yes	37.47	40.45
1-2-3-4	Ladder	Yes	**37.72**	**40.59**
1-2-3-4	Ladder	No	37.49	40.42
1-2-3-4	ResNet	Yes	36.04	38.56

This finding is further confirmed when the same normalization is applied on the training, i.e., when the cosine or combined kernel is used, since in this case the obtained results are further improved.

Furthermore, an ablation study was conducted to evaluate the effectiveness of the proposed multi-layer ladder-based KT. The results are reported in Table X. First, the effectiveness of the multi-layer knowledge transfer is evaluated using one (plain PKT), two, three or four intermediate ladder layers. The precision steadily increases as more layers are employed for the KT process, raising mAP by approximately 0.5% (average relative mAP increase) for each additional layer employed, leading to a total relative accumulative increase of about 1.5%. Furthermore, the effectiveness of the proposed cyclical training process was also evaluated, by simultaneously transferring the knowledge from all the four layers, instead of using the proposed cyclical training process. Indeed, the employed cyclical training process lead to slight, yet consistent, improvements over the baseline. Finally, to further highlight the effectiveness of employing a ladder architecture, multi-layer PKT results are also provided using the ResNet teacher model, instead of the ladder network, where the output of each residual block was matched with each of the layers of the student network. Directly using the larger ResNet teacher, instead of the proposed ladder architecture, leads to a significant drop of the effectiveness of KT, since mAP drops by more than 4% (relative decrease), demonstrating the effectiveness of the proposed ladder-based KT.

V. CONCLUSIONS

In this paper, a novel probabilistic KT method that allows for transferring the knowledge contained in a large and complex neural network into a smaller and faster one was presented. PKT was capable of transferring the knowledge into a smaller student model by keeping as much information as possible, as expressed through the representations extracted from the teacher model. The proposed method is able to employ different kernels to estimate the probability distribution of the teacher and student models, as well as different divergence metrics, allowing for easily adapting to a wide range of different applications. The flexibility of the proposed method was demonstrated using extensive experiments on four different datasets using a wide variety of experimental setups. The robustness and powerful probabilistic formulation of the proposed method led to improved performance in all the evaluated scenarios, overcoming the limitations and significantly outperforming all the evaluated baseline and state-of-the-art KT methods. The experimental results also demonstrated the importance of using the appropriate intermediate layers for KT, since in the case of using a ResNet-18 teacher the efficiency of multi-layer PKT is actually lower than the baseline PKT. To this end, reinforcement learning [46], and/or attention mechanisms [47] can be employed allowing to better select and/or pair the layers between the student and teacher models, further improving the efficiency of KT.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. on Neural Networks and Learning Systems*, 2019.

[3] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2063–2079, 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 4107–4115.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[7] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Proc. Int. Conf. on Learning Representations*, 2015.

[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Neural Information Processing System Deep Learning Workshop*, 2015.

[9] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2016, pp. 5900–5904.

[10] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," in *Proc. Interspeech*, 2016, pp. 3439–3443.

[11] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 742–751.

[12] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop on Similarity-Based Pattern Recognition*, 2015, pp. 84–92.

[13] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 2008.

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

[15] P. Chitrakar, C. Zhang, G. Warner, and X. Liao, "Social media image retrieval using distilled convolutional neural network for suspicious e-crime and terrorist account detection," in *Proc. IEEE Int. Symposium on Multimedia*, 2016, pp. 493–498.

[16] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. Journal of Computer Vision*, vol. 113, no. 1, pp. 54–66, 2015.

[17] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. ACM Conf. on Computer and Communications Security*, 2015, pp. 1310–1321.

[18] R. Fierimonte, S. Scardapane, A. Uncini, and M. Panella, "Fully decentralized semi-supervised learning via privacy-preserving matrix completion," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2699–2711, 2016.

[19] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[20] N. Passalis and A. Tefas, "Unsupervised knowledge transfer using similarity embeddings," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 946–950, 2018.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[22] K. Torkkola, "Feature extraction by non-parametric mutual information maximization," *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1415–1438, 2003.

[23] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proc. European Conf. on Computer Vision*, 2018, pp. 268–284.

[24] L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 2907–2916.

[25] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. Conf. on Artificial Intelligence and Statistics*, 2011, pp. 215–223.

[26] 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," *Int. Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[27] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining*, 2006, pp. 535–541.

[28] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. on Computer Vision*, 2015, pp. 4068–4076.

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

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

[31] A. K. Balan, V. Rathod, K. P. Murphy, and M. Welling, "Bayesian dark knowledge," in *Proc. Advances in Neural Information Processing Systems*, 2015, pp. 3438–3446.

[32] X. Dong, J. Shen, L. Shao, and M.-H. Yang, "Interactive cosegmentation using global and local energy optimization," *IEEE Trans. on Image Processing*, vol. 24, no. 11, pp. 3966–3977, 2015.

[33] J. Shen, Z. Liang, J. Liu, H. Sun, L. Shao, and D. Tao, "Multiobject tracking by submodular optimization," *IEEE Trans. on Cybernetics*, vol. 49, no. 6, pp. 1990–2001, 2018.

[34] J. Shen, X. Dong, J. Peng, X. Jin, L. Shao, and F. Porikli, "Submodular function optimization for motion clustering and image segmentation," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2637–2649, 2019.

[35] J. Shen, J. Peng, X. Dong, L. Shao, and F. Porikli, "Higher order energies for image segmentation," *IEEE Trans. on Image Processing*, vol. 26, no. 10, pp. 4911–4922, 2017.

[36] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proc. European Conf. on Computer Vision*, 2018, pp. 459–474.

[37] S.-I. Mirzadeh, M. Farajtabar, A. Li, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher," *arXiv preprint arXiv:1902.03393*, 2019.

[38] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 7130–7138.

[39] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[40] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[41] B. A. Turlach *et al.*, *Bandwidth selection in kernel density estimation: A review*. Université catholique de Louvain Louvain-la-Neuve, 1993.

[42] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proc. Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.

[43] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical Report*, 2009.

[44] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2012, pp. 2751–2758.

[45] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for k-means clustering," in *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, 2009, pp. 877–886.

[46] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, and F. Porikli, "Hyperparameter optimization for tracking with continuous deep q-learning," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 518–527.

[47] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Trans. on Image Processing*, vol. 27, no. 5, pp. 2368–2378, 2017.

Nikolaos Passalis received the B.Sc. in Informatics in 2013, the M.Sc. in Information Systems in 2015 and the Ph.D. degree in Informatics in 2018, from the Aristotle University of Thessaloniki, Greece. Since 2019 he has been a post-doctoral researcher at the Aristotle University of Thessaloniki, Greece. He has (co-)authored more than 55 journal and conference papers and contributed one chapter to one edited book. His research interests include deep learning, lightweight machine learning, information retrieval and computational intelligence.

Maria Tzelepi received the B.Sc. degree in informatics and the M.Sc. degree in digital media-computational intelligence, both from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2013 and 2016, respectively. She is currently working toward the Ph.D. degree with the Artificial Intelligence and Information Analysis Laboratory, Department of Informatics, Aristotle University of Thessaloniki. Her research interests include deep learning, pattern recognition, computer vision, and image analysis and retrieval.

Anastasios Tefas received the B.Sc. in informatics in 1997 and the Ph.D. degree in informatics in 2002, both from the Aristotle University of Thessaloniki, Greece. Since 2017 he has been an Associate Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 2008 to 2017, he was a Lecturer, Assistant Professor at the same University. From 2006 to 2008, he was an Assistant Professor at the Department of Information Management, Technological Institute of Kavala. From 2003 to 2004, he was a temporary lecturer in the Department of Informatics, University of Thessaloniki. From 1997 to 2002, he was a researcher and teaching assistant in the Department of Informatics, University of Thessaloniki. Dr. Tefas participated in 18 research projects financed by national and European funds. He is Area Editor in Signal Processing: Image Communications journal. He has co-authored 100 journal papers, 225 papers in international conferences and contributed 8 chapters to edited books in his area of expertise. Over 5000 citations have been recorded to his publications and his H-index is 36 according to Google scholar. His current research interests include computational intelligence, deep learning, pattern recognition, statistical machine learning, digital signal and image analysis and retrieval and computer vision.