ROBUST HYPERSPHERE-BASED WEIGHT IMPRINTING FOR FEW-SHOT LEARNING

Nikolaos Passalis[1], Alexandros Iosifidis[2], Moncef Gabbouj[3] and Anastasios Tefas[1]
[1]Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
[2]Department of Engineering, Electrical and Computer Engineering, Aarhus University, Denmark
[3]Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland
E-mails: passalis@csd.auth.gr, ai@eng.au.dk, moncef.gabbouj@tuni.fi, tefas@csd.auth.gr

***Abstract*—Performing fast few-shot learning is increasingly important in a number of embedded applications. Among them, a form of gradient-descent free learning known as Weight Imprinting was recently established as an efficient way to perform few-shot learning on Deep Learning (DL) accelerators that do no support back-propagation, such as Edge Tensor Processing Units (Edge TPUs). Despite its efficiency, WI comes with a number of critical limitations. For example, WI cannot effectively handle multimodal novel categories, while it is especially prone to overfitting that can have devastating effects on the accuracy of the models on novel categorizes. To overcome these limitations, in this paper we propose a robust hypersphere-based WI approach that allows for regularizing the training process in an imprinting-aware way. At the same time, the proposed formulation provides a natural way to handle multimodal novel categories. Indeed, as demonstrated through the conducted experiments, the proposed method leads to significant improvements over the baseline WI approach.**

***Index Terms*—Weight Imprinting, Few-shot Learning, Edge TPU, Embedded Deep Learning**

I. Introduction

Deep Learning (DL) has achieved remarkable results on a wide range of especially challenging problems [1], which span from image understanding and visual questioning answering problems [2], to natural language processing [3], and video analysis [4]. Despite the success of DL on the aforementioned areas, DL models require powerful hardware both during the training process, as well as during the inference. The development of advanced Graphics Processing Units (GPUs) [5], as well as Tensor Processing Units (TPUs) [6], allowed for accelerating both the training and inference processes, as well as lowering the energy requirements of DL. However, most of these accelerators remain too bulky for many embedded applications, including robotics [7] and Internet-of-Things (IoT) applications [8], leading to developing extremely energy efficient hardware for accelerating only the *inference* process, such as Edge TPUs [9]. These platforms can lead to tremendous improvements in terms of operations/Watt. However, being designed to support inference-only operations, they are unable to support back-propagation, rendering impossible to perform on-device training. These limitations severely restrict their applications on open-word settings, which frequently occur in many applications, e.g., robotics [10], where DL models should be able to promptly adapt to emerging categories that were not seen during the training. This problem is known as *few-shot learning* or *low-shot learning* [11], [12], [13], where models are required to learn new categories and generalize the already encoded knowledge using just a few labeled examples.

Despite the large number of few-shot learning methods proposed in the literature [11], [12], [13], [14], just a few of them support inference-only DL accelerators. Perhaps the most widely known is the *Weight Imprinting* (WI) approach [15], which was recently proposed as the default way to perform training on the Google's Edge TPU accelerator. WI allows for extending the classes a DL model can recognize by performing gradient descent-free learning. To this end, an appropriate vector prototype is calculated for each novel class and, then, this prototype vector is *imprinted* in the last classification layer of a DL model. Note that this process does not require back-propagating any gradients through the network, while the required calculations can be efficiently performed using an embedded CPU, rending the whole process extremely efficient.

However, WI suffers from significant limitations, that hinders its application in many real-world cases. First of all, WI implicitly assumes that the distribution of the features extracted from the DL model for novel categories will be unimodal. While this assumption usually holds for the distribution of classes for which the DL model was trained, there is no guarantee that this will hold for novel categories, which were never presented to the network during the training process. Furthermore, the impact of the training process on the actual performance on the network during few-shot learning has not been adequately examined in existing literature. More specifically, as we also experimentally demonstrate in this paper, there seems to be a direct connection between maintaining the variance of the embeddings around the prototypes and the generalization abilities of a representation/model on unknown classes. This is not a surprising result, since it is well known that overfitted representations almost always lead to worse generalization (after a certain point) [16], [17], [18]. This naturally leads us to the following question: Is it possible to design a representation in which the variance around the prototypes will be deliberately controlled to achieve the perfect balance between overfitting and underfitting instead of relying on early-stopping, implicit regularization or other heuristics to maintain enough variance? Also note that maintaining the variance will allow more information about the in-class similarities/dissimilarities to be encoded in the resulting

representation.

The main contribution of this paper is a novel weight imprinting method, which is capable of overcoming the aforementioned limitations. To this end, the proposed method learns a regularized embedding by maintaining the variance around the prototypes in a structured way. The proposed approach provides an effective a way to directly handle novel categories with multimodal distributions, as well as natively supports few-shot learning. To the best of our knowledge, in this paper we propose the first variance-preserving approach for performing imprinting-aware few-shot learning.

The rest of this paper is structured as follows. First, the proposed method is derived and discussed in Section II. Then, the proposed method is evaluated and compared to regular WI under different scenarios in Section III. Finally, conclusions are drawn and future research directions are discussed in Section IV.

II. Proposed Method

The proposed method aims to learn a carefully designed feature space to more effectively support weight imprinting. To this end, a centroid-based loss, which uniformly distributes the embedding vectors within a radius r around each prototype (centroid), is employed. Furthermore, to ensure that the prototypes are discriminative enough it is required that the minimum distance between two prototypes is at least $\rho > 2r$. This process is illustrated in Fig. 1. After learning a representation that fulfills the aforementioned requirements we can directly classify a new sample, perform gradient descent-free few-shot learning, detect and handle multimodal novel classes and detect intrusion to the existing classes that can lower the performance of the model.

A. Embedding Extractor Training

First, we will describe the proposed imprinting-aware training process. Let $\phi(\mathbf{x}) \in \mathbb{R}^m$ be the output of a neural network, where m is the dimensionality of the embeddings extracted from the network, when presented with an input sample $\mathbf{x}$. Also, let $\mathbf{w}_i$ be the prototype vector for the i-th class used during the training process and $\mathcal{X}_i$ be the set of samples that belong to the class i. Then, to ensure that the embeddings will be uniformly distributed around each class prototype $\mathbf{w}_i \in \mathbb{R}^m$ we define the appropriate class-induced loss as:

$$\mathcal{L}_c = \frac{1}{N} \sum_{l=1}^{N_C} \sum_{\mathbf{x} \in \mathcal{X}_l} \left(||\phi(\mathbf{x}) - \mathbf{w}_l||_2 - \alpha r \right)^2, \tag{1}$$

where N is the total number of training samples, N_C is the total number of training classes, $||\cdot||_2$ denotes the l^2 norm of a vector and $\alpha \in [0, 1]$ is a number drawn uniformly from the range $[0...1]$. During the optimization a different random value is drawn for α for each sample and iteration, leading to a uniform distribution of the embeddings within a radius r from each $\mathbf{w}_i$. Even though this process does not ensure that the full space around $\mathbf{w}_i$ will be occupied, it ensures that the embeddings will be sampled uniformly at various radiuses around the corresponding center, significantly improving the generalization abilities of the representation, as we will demonstrate later in Section III. Note that by setting $r = 0$, the loss $\mathcal{L}_{class}$ degenerates to the regular center loss [19]. Furthermore, to further model the uncertainty regarding the class prototypes, we can use Gaussian noise to corrupt the prototypes as:

$$\tilde{w}_i = w_i + \mathcal{N}(0, \sigma). \tag{2}$$

At the same time, each prototype $\mathbf{w}_i$ is required to be at a distance of at least ρ from each other prototype (to ensure that there is no overlapping between the hyperspheres that enclose the embeddings of each class). To this end, we also define the prototype loss as:

$$\mathcal{L}_p = \frac{1}{N_C(N_C - 1)} \sum_{i=1}^{N_C} \sum_{j=1, i \neq j}^{N_C} \max(0, \rho - ||\mathbf{w}_i - \mathbf{w}_j||_2). \tag{3}$$

Therefore, the model parameters, along with the prototypes, are optimized during the training to minimize the following loss:

$$\mathcal{L} = \mathcal{L}_c + \gamma \mathcal{L}_p, \tag{4}$$

where γ is a hyper-parameter parameter that alters the weight of the prototype loss (set to 1 for all the experiments conducted in this paper).

B. Classification and Few-shot Learning

To classify an input sample we can directly choose the class that corresponds to the prototype with the smaller distance to the extracted embedding $\phi(\mathbf{x})$. The network can be used in a similar fashion as a one trained using the softmax activation simply by using a final classification layer that calculates the membership value for each prototype/class probabilities as:

$$p_i(\mathbf{x}) = \frac{1}{1 + ||\phi(\mathbf{x}) - \mathbf{w}_i||_2}. \tag{5}$$

It is worth noting that the probability distribution can be smoothed or sharpened by appropriately transforming it, e.g., using the softmax function with appropriately tuned temperature, if needed.

To perform few-shot learning we can simply augment the final classification layer with an additional prototype vector $\mathbf{w}_n$ calculated as:

$$\mathbf{w}_n = \frac{1}{|\mathcal{X}_n|} \sum_{\mathbf{x} \in \mathcal{X}_n} \phi(\mathbf{x}), \tag{6}$$

where $\mathcal{X}_n$ is the set that contains the training samples for the novel category. Note that similarly to regular WI, no gradient descent-based optimization is required for extending the classifier to support novel classes. However, as we further demonstrate in Section III, the regularized nature of the learned feature space leads to significantly better performance compared to regular WI.

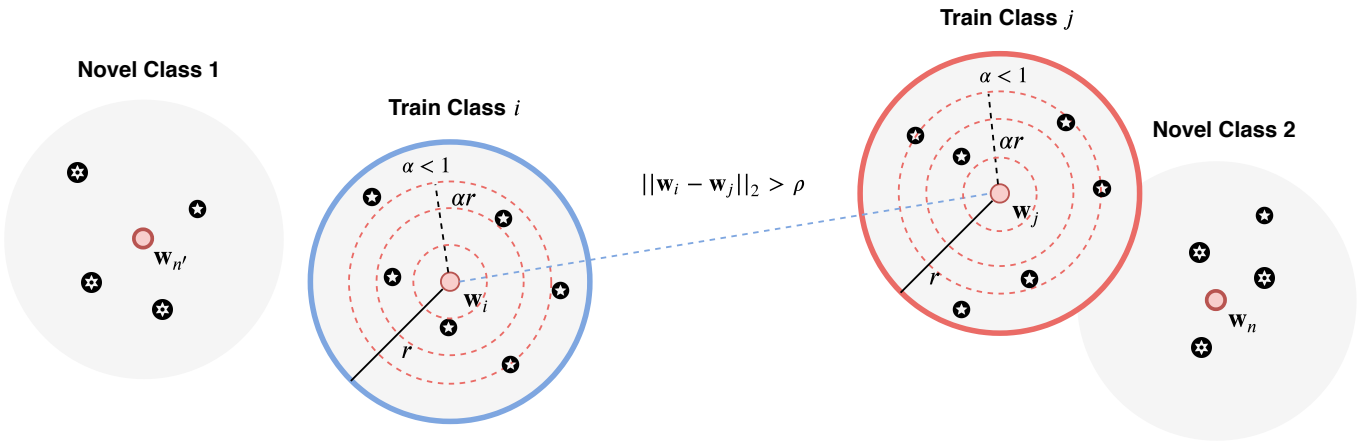

Fig. 1. Weight Imprinting using Hyperspheres: The representation space is constructed in a way that natively supports imprinting and spreads the embeddings in hyperspheres with radius r (the in-class variance is better preserved). Note that the proposed method also allow for detecting when the imprinting process cannot be performed safely (e.g., note the potential overlap between the second novel class and the j-th class).

C. Detecting and handling multimodal novel categories

There are several ways to detect if the distribution of a novel class is indeed multimodal. In this paper, we propose using a quite simple, yet effective way: we propose directly detecting multimodal categories by clustering the embedding vectors extracted for a novel category. If we detect centers that are at distance greater than r (or any other user-defined threshold) from each other, then a hypersphere with radius of r cannot enclose the embeddings of the novel class. To address this, we can simply add one or more prototypes (according to the number of centers that are at distance greater than r) to model the distribution of the novel class. In this way, one class can be represented using more than one prototype. On the other hand, if the centers of the clusters are within a radius of r, then we assume that proposed classification scheme can directly handle the distribution of the novel class (even though there is no guarantee that the distribution is not multimodal).

III. Experimental Evaluation

The proposed method was evaluated using two different image datasets, the MNIST dataset [20] and the Animals with Attributes (AwA2) dataset [21], in order to measure its performance under two different settings. For the MNIST dataset the first 5 classes were used to train the model, while the remaining 5 classes were used for few-shot learning and evaluation. For the AwA2 dataset [21] the first 40 classes were used for training the model, and the rest of the classes were employed for evaluating the performance of the proposed method. The performance evaluation was repeated 5 times using different training samples for the novel classes (except for the hyper-parameter evaluation experiments) and the mean and standard deviation is reported. For the MNIST dataset the employed neural network was composed of a 3×3 convolutional layer with 32 filters, followed by a 2×2 max pooling layer, another 3×3 convolutional layer with 64 filters, an additional 2×2 max pooling layer and a fully connected layer with 256 neurons. For the AwA2 dataset we used a pretrained ResNet101 to extract feature vectors (as described in [21]) that were then fed to two fully connected layers with 2048 and 512 neurons respectively. The *relu* activation function was used for all the layers [22]. The models were trained using the Adam optimizer with a learning rate of 10^{-3} for 20 training epochs for the MNIST dataset and of 10^{-4} for 20 training epochs for the AwA2 dataset [23].

First, we evaluated the effect of altering the minimum distance ρ between the different class prototypes $\mathbf{w}_i$, while keeping the radius fixed to $r = 0$. The results are presented in Table I. Increasing the minimum distance between the class prototypes seems to have a positive effect on the classification accuracy, both for the novel split (denoted by "Novel") and the combined split of novel and training classes (denoted by "All"). This was expected since the learned representation is not capable of perfectly collapsing the embeddings to the corresponding prototypes, even though the radius was set to $r = 0$. Therefore, keeping a quite large margin between the different prototypes helps to reduce the risk of wrongly classifying a sample.

Next, we also evaluated the effect of altering the radius r on the learned representation. In Section II it was conjectured that spreading the training embeddings in a hypersphere of radius r will have a positive regularization effect on the learned representation by allowing the model to capture and better model the in-class variations. Indeed, as demonstrated in Table II, the classification accuracy increases by more than 14% for the novel split, and by 1.8% for the combined split. This confirms our hypothesis that collapsing the embeddings to the class centers, without any form of regularization, can significantly reduce the classification accuracy, especially when dealing with classes that were not seen during the training process and using powerful models that can overfit the training data.

Next, we evaluated the proposed method using a 1-shot, 2-

TABLE I
EVALUATING THE EFFECT OF THE HYPER-PARAMETERS ρ ON THE 2-SHOT LEARNING ACCURACY OF THE PROPOSED IMPRINTING METHOD ON BOTH THE NOVEL CATEGORIES ("NOVEL" SPLIT) AND THE COMBINED NOVEL AND TRAINING CATEGORIES ("ALL" SPLIT).

Min. Distance ρ	Novel	All
1	56.28	77.79
2	56.24	77.42
5	58.53	78.66
10	**61.02**	**80.06**
20	53.82	76.51

TABLE II
EVALUATING THE EFFECT OF THE HYPER-PARAMETERS r ON THE 2-SHOT LEARNING ACCURACY OF THE PROPOSED IMPRINTING METHOD ON BOTH THE NOVEL CATEGORIES ("NOVEL" SPLIT) AND THE COMBINED NOVEL AND TRAINING CATEGORIES ("ALL" SPLIT).

Radius r	Novel	All
0	61.02	80.06
4	64.51	80.59
5	**69.57**	**81.55**
6	66.86	78.26

TABLE III
MNIST: EVALUATING THE ACCURACY OF IMPRINTING METHODS ON BOTH THE NOVEL CATEGORIES ("NOVEL" SPLIT) AND THE COMBINED NOVEL AND TRAINING CATEGORIES ("ALL" SPLIT).

Method	Split	1-shot	2-shot	5-shot
WI	Novel	52.44 ± 7.4	66.95 ± 4.3	72.51 ± 2.6
Proposed-	Novel	47.46 ± 9.9	59.98 ± 2.6	66.60 ± 3.7
Proposed	Novel	$\mathbf{57.93 \pm 4.2}$	$\mathbf{68.72 \pm 5.1}$	$\mathbf{75.20 \pm 3.5}$
WI	All	57.51 ± 5.8	58.35 ± 6.4	64.49 ± 3.2
Proposed-	All	73.27 ± 4.7	$\mathbf{79.46 \pm 1.5}$	82.88 ± 1.8
Proposed	All	$\mathbf{73.66 \pm 3.9}$	79.09 ± 6.0	$\mathbf{84.23 \pm 1.3}$

TABLE IV
MNIST MULTIMODAL: EVALUATING THE ACCURACY OF IMPRINTING METHODS ON THE NOVEL CATEGORIES.

Method	Thres.	2-shot	4-shot	10-shot
WI	-	55.81 ± 11.7	65.17 ± 12.3	78.89 ± 2.6
Proposed-	-	42.07 ± 6.5	47.40 ± 9.6	55.47 ± 2.4
Proposed	-	60.95 ± 4.0	71.03 ± 5.5	75.69 ± 1.0
Proposed	5	62.36 ± 3.4	70.00 ± 5.3	75.69 ± 1.0
Proposed	4	66.47 ± 5.0	71.52 ± 2.3	75.11 ± 2.8
Proposed	3	$\mathbf{66.78 \pm 4.9}$	$\mathbf{73.99 \pm 5.6}$	$\mathbf{82.69 \pm 1.5}$

shot and 5-shot evaluation protocol on the MNIST dataset. The results are reported in Table III. Two variants of the proposed method were evaluated: "Proposed-", where $r = 0$ and $\sigma = 0$ were used, and "Proposed", where $r = 5$ and $\sigma = 0.05$ were used. The proposed method was also compared to the plain Weight Imprinting (WI) approach [15], using an initial scaling value of $c = 10$. Again, it was confirmed that using the proposed variance preserving approach improves the performance over simply using a center-based loss, allowing for outperforming the regular WI method on all the evaluation splits and few-shot learning setups. It is worth noting that the accuracy for all the evaluated methods is relatively low compared to the state-of-the-art, since neither WI or the proposed method perform any kind of optimization according to a discriminative objective.

To demonstrate the ability of the proposed method to handle multimodal classes we also evaluated the proposed approach using an additional multimodal split of the MNIST dataset. This split was compiled by merging two succeeding classes into one, e.g., "0" and "1" were merged into a new class, "2" and "3" into another, and so on. Then, the three first classes (digits 0 to 5) were used for training and the remaining two of them (6 to 9) for evaluating the few-shot learning performance. The evaluation results are reported in Table IV. The employed threshold was used to detecting whether a class distribution is multimodal (by clustering the training data into two clusters and measuring the distance between the resulting centroids). If the distance of the resulting centers was greater than the specified threshold, then two prototypes were used per novel class. Again, note that the variance-preserving variant of the proposed method greatly outperforms the baseline variant ("Proposed-"), as well that using the proposed multimodal handling approach (with a correctly tuned threshold) allows for significantly improving the performance of the proposed method, outperforming all the other evaluated approaches.

Finally, we also evaluated the performance of the proposed method using a more challenging dataset, the AwA2 dataset. The results are reported in Table V. We report evaluation results only for the novel categories, since due to the small number of training samples, all the data from the training categories were used for training the model. As before, the proposed method leads to significant performance improvements over the plain WI method, while it still outperforms the HWI-methods. The smaller differences between HWI- and HWI can be possibly attributed to the the smaller learning capacity of the employed network (the risk of overfitting the representation is higher when more powerful networks are employed). Note that slightly different hyper-parameters were used for the HWI method in this experiment: $\rho = 20$, $r = 10$, and $\sigma = 0$.

IV. CONCLUSIONS

In this paper we proposed a novel hypersphere-based weight imprinting approach that maintains all the advantages of regu-

TABLE V
AWA2: EVALUATING THE ACCURACY OF IMPRINTING METHODS ON THE COMBINED NOVEL AND TRAINING CATEGORIES SPLIT.

Method	1-shot	2-shot	5-shot
WI	51.03 ± 3.71	61.33 ± 2.73	75.23 ± 1.85
Proposed-	54.55 ± 3.31	68.44 ± 3.17	76.95 ± 2.12
Proposed	$\mathbf{56.14 \pm 2.70}$	$\mathbf{70.16 \pm 2.63}$	$\mathbf{77.85 \pm 1.84}$

lar WI [15], i.e., it is able to readily extent a pretrained neural network to classify samples from novel categories simply by adding new weight vectors in the final classification layer without requiring to perform any form of back-propagation to this end. At the same time, the proposed method was capable to overcome significant limitations of WI by being able to learn regularized representations that provide better generalization for classes which were not seen during the training and provide a straightforward way to directly handle novel categories with multimodal distributions. The proposed method was extensively evaluated on two image datasets, outperforming the regular WI approach. However, it is worth noting that the imprinting process is still quite behind traditional gradient descent-based learning approaches.

There are several interesting future research directions. First, even though the proposed clustering-based multi-modality detection led to adequate results, several most sophisticated methods can be used for detecting whether the distribution of a novel class is indeed multimodal. Also, to better spread the in-class samples in the volume of the hypersphere we employed a stochastic process that randomly distributes the samples in various radiuses. However, this process ignores the actual geometry of the data. More advanced methods that take into account the actual manifold of the data, e.g., by using data from a previous layer and then distilling the extracted information into the used model [24], [25], [26], can be employed. Finally, the proposed hypersphere-based loss showed a great potential for improving regular training tasks as well. Therefore, it can be potentially successfully used even for regular classification tasks, further improving the accuracy of the corresponding models.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449 (OpenDR). 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] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[2] Huijuan Xu and Kate Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 451–466.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 20–36.

[5] Nicolas Vasilache, Jeff Johnson, Michael Mathieu, Soumith Chintala, Serkan Piantino, and Yann LeCun, "Fast convolutional nets with fbfft: A gpu performance evaluation," *arXiv preprint arXiv:1412.7580*, 2014.

[6] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al., "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the Annual International Symposium on Computer Architecture*, 2017, pp. 1–12.

[7] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al., "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[8] He Li, Kaoru Ota, and Mianxiong Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[9] Zhiming Hu, Ahmad Bisher Tarakji, Vishal Raheja, Caleb Phillips, Teng Wang, and Iqbal Mohomed, "Deephome: Distributed inference with heterogeneous devices in the edge," in *Proceedings of the International Workshop on Deep Learning for Mobile Systems and Applications*, 2019, pp. 13–18.

[10] Stephen James, Michael Bloesch, and Andrew J Davison, "Task-embedded control networks for few-shot imitation learning," *arXiv preprint arXiv:1810.03237*, 2018.

[11] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2011, vol. 33.

[12] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *Proceedings of the Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[13] Bharath Hariharan and Ross Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proceedings of the Int. Conference on Computer Vision*, 2017, pp. 3018–3027.

[14] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan, "Low-shot learning from imaginary data," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7278–7286.

[15] Hang Qi, Matthew Brown, and David G Lowe, "Low-shot learning with imprinted weights," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5822–5830.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] Nikolaos Passalis and Anastasios 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.

[18] Nikolaos Passalis, Alexandros Iosifidis, Moncef Gabbouj, and Anastasios Tefas, "Variance-preserving deep metric learning for content-based image retrieval," *Pattern Recognition Letters*, vol. 131, pp. 8–14, 2020.

[19] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 499–515.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata, "Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2018.

[22] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[25] Nikolaos Passalis and Anastasios Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 268–284.

[26] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2339–2348.