Exploring Deep Learning Techniques for Low-resource Image Classification

(s2238419, s2259832, s2247978, s2101601)

Abstract

Deep learning has helped many applications of computer vision reach its state-of-the-art performance. The recent progress in computer vision has been because of availability of denser models and large-scale datasets. However, for the fairly unexplored domains, small datasets poses as a limitation to the otherwise effective deep learning solutions. Researchers proposed the use of transfer learning and knowledge distillation techniques to take advantage of large-scale data of a similar domain. But, it assume availability of such large-scale data. In this work we focus on the Image Classification track of the VIPrior challenge that simulates availability of a limited dataset under constraints that prohibit techniques that allow knowledge transfer from large-scale dataset of the similar domain. We take one step ahead and put constrained on the model architecture we can use, i.e., the task is limited to using ResNet-50 (not pre-trained). We study different well known techniques and apply it to the image classification dataset. We find that among the explored augmentation techniques, MixUp gives improves the test performance by 1.13%. When compared against the baseline, REPTILE, a meta-learning approach, achieved slightly better performance on the test set. Under the similarity learning paradigm, Siamese learning with contrastive loss performs the best on test set and outperforms the baseline on the validation. We also explored graph neural networks under a special training setting to explore the contextual relationships between different samples.

1. Introduction

Deep learning is fueling the success of many recent advances in computer vision and machine learning (LeCun et al., 2015). The contributing factors to this success of deep learning is (i) its deeper network architecture, and (ii) use of large training datasets. However, obtaining large annotated datasets is expensive, limiting many applications to exploit deep learning to its full potential. To mitigate this limitation, researchers birthed a new field of deep learning in a limited data setting. Bengio (2012) introduced transfer learning, a clever way to learn complex representations from one domain and transfer that knowledge (learned features) to a closely related domain (Pan & Yang, 2010). Based on the similar concept, Vanschoren (2018) proposed few-shot learning which is capable of generalizing scarce support set target classes using the label pairs in the base set. Knowledge distillation (Hinton et al., 2015; Romero et al., 2014; Zagoruyko & Komodakis, 2016), another popular technique trains a smaller network (student network) under the supervision of the larger network (teacher network), with the goal of improving the performance of the student network. These techniques assume that large datasets of the similar domain as the target dataset are accessible.

To advance methodological advances, research communities started challenges like VIPriors (visual inductive priors for data-efficient deep learning) (Bruintjes et al.) which is designed for applications with limited or small datasets. It additionally constraints the participants from using pre-trained checkpoints and knowledge-transfer techniques like the ones discussed in the prior paragraph. We focus on the image classification track of this challenge. During its course, participants proposed working with data augmentation techniques, and different architectural modifications to improve the classification accuracy. The first place winners proposed working with the Dual Selective Kernel network (DSKnet) (Sun, 2021) with three losses - positive class loss, center loss (Wen et al., 2016) and tree supervision loss inspired by Wan et al.. They also performed CutMix (Yun et al., 2019a) data augmentation technique. Pengfei Sun proposed working with mixture of experts model to learn enriched representations by using different neural architectures combined via same initial backbone layers. They augmented their data using performed data augmentation techniques like AutoAugment (Cubuk et al., 2018), MixUp (Zhang et al., 2017a), and CutMix (Yun et al., 2019a). The second place winners proposed, Luo trains an emsemble of ResNest-101 (Zhang et al., 2020), TresNet-XL (Ridnik et al., 2021), Rexnet (Han et al., 2021) and an Inception-ResNet (Szegedy et al., 2017), using ”dynamic semantic scale balance loss” (DSB) (Luo et al., 2021). The third place winners, Kim et al. (2020) train EfficientNet (Tan & Le, 2019) and augment data using Low Significance Bit swapping between image pixels. They use focal (Lin et al., 2017) and cosine loss (Barz & Denzler, 2019) to train their model. Tan Wang proposed Iterative Partition-based Invariant Risk Minimization, a self-supervised learning technique to disentangle the different semantic concepts of a representation. Their proposed solution involves knowledge distillation from the teacher model to the student model. During knowledge distillation they augmented their data using RandAugment (Cubuk et al., 2019) and AutoAugment (Cubuk et al., 2018) techniques. The success of above men-

tioned works is largely governed by the complex ensemble models they proposed and their training techniques (loss functions).

In this paper, we work on the VIPrior challenge under some additional constraints. Besides the rules of VIPrior challenge that prohibits the participants from using additional data and pre-trained methods, we are also constrained by the choice of the model architecture, i.e., limited to using ResNet-50. Under these conditions we study different methods - data augmentation techniques, and different training settings. Inspired by the works of winning solutions of the challenge, we explore data augmentation techniques namely CutMix (Yun et al., 2019b) and MixUp (Zhang et al., 2017b). Owning to the success of similarity based learning in low data setting, we explore three type of similarity leaning techniques - (i) Siamese network (Utkin et al., 2021) using contrastive loss and the triplet loss, (ii) supervised contrastive learning (Khosla et al., 2020), and learning with cosine similarity (Barz & Denzler, 2020). We found that Siamese network trained via contrastive loss worked the best for VIPrior data among the discussed similarity learning techniques. We also trained REPTILE (Nichol et al., 2018), a model agnostic meta learning approach to explore few-shot learning which acheived slightly better performance on the test set than the baseline. Additionally, we train a Graph Neural Network (Zhou et al., 2020) under a special setting to explore the contextual relationships.

2. Dataset

In this paper we work with the dataset provided by the VIPriors Image Classification Challenge[1]. It is a subset of ImageNet-1k (Russakovsky et al., 2014), a dataset of 1,000 classes of everyday objects and natural scenes. Each 1,000 class in the VIPrior challenge dataset has exact 50 images per class. The dataset is split into 50,000 training, 50,000 validation images, and 100,000 test images. The task is to perform image classification with limited data, without using transfer learning and pre-training methods. But in our paper, we use the modified data with reduced 50,000 training dataset out of which 20% data is kept separate as validation data. Test set is also reduced to 50,000 images.

3. Methodology

3.1. Data Augmentation

One of the ways to increase the training data for the classifier would be to generate images artificially using data augmentation techniques on the provided data. Data augmentation is a method in which we make small changes to the data to create new data points. Using these techniques, we create multiple representations of an image which are then included in our training data. Some of the most commonly used augmentations are rotating, flipping, zooming etc. However, after experimenting with the above mentioned common techniques, we found the complex augmentation techniques like CutMix and Mixup to be more effective.

3.1.1. Mixup

In (Zhang et al., 2017b), the authors introduce the mixup augmentation where a new image is formed using weighted linear relation between two images that are already present in the training data.

$$\begin{aligned} \tilde{x} &= \lambda x_A + (1-\lambda) x_B \\ \tilde{y} &= \lambda y_A + (1-\lambda) y_B \end{aligned} \tag{1}$$

In eq 1, (x_A, y_A) and (x_B, y_B) are two images with x being the input vector and y being the label of these images.

Figure 1. Mixup of a Cat and a Dog image

In Figure 1, we can see how the equations in 1 are used to create a new image combining a dog and a cat. In eq 1 y_A and y_B are the true labels for both the images and $\tilde{y}$ is the generated combination of labels based on λ. We sample λ based on a beta distribution ($Beta(\alpha, \alpha)$).
One of the main advantages of using mixup is that it allows linear transition of the decision boundaries between classes which provides us a smoother estimate of uncertainty in the prediction of a classifier.

3.1.2. CutMix

In the CutMix augmentation (Yun et al., 2019b), we replace parts of image with a patch from another image that is already present in the training data (Figure 2). The labels are also then changed to a proportion between the labels of these two images which are used in the CutMix strategy. This proportion is based on the number of pixels covered by each image in the resulting image.

$$\tilde{x} = M \cdot x_A + (1 - M) \cdot x_B \tag{2}$$

Where $\tilde{x}$ is the final image, M is a binary mask denoting the cutout and fill in regions of the two images and x_A and x_B are the two images that we want to combine.

$$\tilde{y} = \lambda y_A + (1-\lambda) y_B \tag{3}$$

In eq 3, $\tilde{y}$ is generated similarly as we generate that in mixup. However, here, we sample λ from a uniform distribution (0,1) which is based on a beta distribution (where $\alpha = 1$).

3.2. Similarity Learning

Similarity Learning is a Machine Learning paradigm where inputs are mapped into a high dimensional space, so that

[1] https://github.com/VIPriors/vipriors-challenges-toolkit/tree/master/image-classification

Figure 2. CutMix of a Cat and a Dog image

similar inputs are located at nearby position in the new space and different inputs far away from each other. Similarity Learning is usually used as a representation learning to learn efficient representation of inputs which can be used later effectively for further downstream tasks. We have explored three different techniques in this domain.

3.2.1. Siamese Networks

Siamese networks (Utkin et al., 2021) consists of two or more parallel networks trained with shared weights. These parallel networks are also called Sister networks or Tower networks. It learns how to generate the effective representation to maximise the separation between two dissimilar inputs and minimise the separation between two similar inputs. There are several training settings possible for Siamese (Suprapto & Polela, 2020), we have explored with contrastive and triplet loss Fig. 4.

Siamese with contrastive loss

This is a two stage learning process. At first stage, input dataset consists of pairs of images which could be either of same class and labelled as 1 or could be of different class and labelled as 0. Once we have created this pair dataset, it is passed through a embedding network and then it was trained with the contrastive loss with this binary output. After training it with contrastive loss (Wang & Liu, 2021) embedding network can be used to generate the new embeddings for the image which is then later trained with simple Multi Layer Perceptron (MLP) as our final classification model.

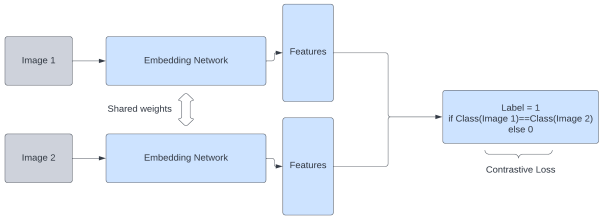

Figure 3. Siamese with contrastive loss network architecture

Siamese with triplet loss

Siamese with triplet loss consists of three inputs anchor, positive and negative. Anchor refers to the original data point, positive refers to the random data point of same class and negative refers to random data point of different class. Each of these inputs are passed through same embedding networks and final layer is concatenated into a single vector and then the triplet loss is applied (Wang & Liu, 2021). After we have trained this network, we used the embedding network for new image embeddings which is finally trained with a simple MLP.

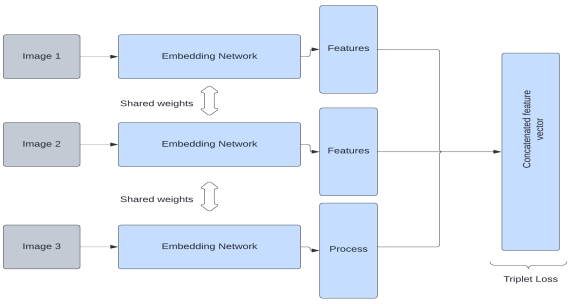

Figure 4. Siamese with triplet loss network architecture

3.2.2. Supervised Contrastive

(Khosla et al., 2020) propose a new way of pretraining a network for a classification task, where we use supervised contrastive loss for pretraining and cross entropy loss for training the classifier. This method aims at learning better representations first such that the classifying them would become easier. The cross entropy loss only learns to separate the samples using the decision boundary, i.e it only cares the samples to be on one or the other side of the of the decision boundary. However, using supervised contrastive loss we pull the representations from same classes together and push away the representation of other classes. After learning these representations, we freeze the whole network and just train the classifier on the cross entropy loss to have better decision boundaries as now representations from similar classes will be pulled together because of the supervised contrastive learning.

3.2.3. Cosine Similarity

Cosine Similarity (Barz & Denzler, 2020) method aims at increasing the cosine similarity between the output of neural net and the target class one hot by minimizing the cosine loss. Cosine loss substitute the MSE and categorical cross entropy loss which are generally used in learning a neural classifiers. Cosine loss values are bounded in the range of [0,2] unlike cross entropy which can take arbitrarily large value. Cosine loss depends only on the direction of features and is invariant to the magnitude of feature scaling. This makes cosine loss more robust to be applied across datasets with varying number of classes. In our experiment, cosine loss does not perform as well as other similarity learning methods and has very low accuracy on both validation and test dataset. In its paper (Barz & Denzler, 2020) took five different datasets, each of which pertains to very specific type of images like birds, cars, flowers etc. Our dataset has quite wide variety of image classes and could lead to important changes on how model preforms.

3.3. Meta Learning

Meta Learning comprises of creating tasks that has an N-way K-shot classification where N is the number of classes and K is the number of instances of each class in training set. These sets are know as support set which contains randomly sampled classes and used to perform learning task. The model is evaluated on query set of the same sample and accordingly the model parameters are updated. The model is trained iteratively by following the sampling process in each iteration.
Meta Learning is further divided on basis of types of prior knowledge (W Zi, 2019)- prior knowledge about similarity, prior knowledge about learning and prior knowledge about data. In this paper, we focus on Reptile (Nichol et al., 2018), a kind of first-order MAML that depends on prior knowledge about learning.

3.3.1. REPTILE

Model Agnostic Meta learning (MAML) (Finn et al., 2017) is a optimization based few shot learning technique. This technique uses a specific learning procedure that enable it to learn on limited input data. A general purpose algorithm is deployed with any model that leverages Stochastic Gradient Descent (SGD) for learning. MAML emphasises on creating a copy of initialisation weights and running one iteration of gradient descent for a random task on the copy. Loss function calculated on the query set is used to update the initial weights in each iteration. This method however requires computation of second order gradients which makes learning process slower. For a defined model with a random function f_ϕ with parameters ϕ, a task τ is randomly sampled from dataset, the weights are updated in each iteration using gradient descent by minimising the loss function L as given by eq. 4

$$\phi^{'} \rightarrow \phi - \nabla L_\tau(f_\phi) \tag{4}$$

Reptile (Nichol et al., 2018) is a form of one first-order MAML that does not involve second order derivatives and try to generalise network from a small batch of gradients from query task. This allows for faster learning without compromising on performance. After performing the SGD for each iteration as given in eq 4, $(\phi - \phi^{'})$ is treated as a gradient as depicted in eq 5. and used to move the parameter values more towards the new set $\phi^{'}$. Eventually ϕ will converge towards optimal value for all task sets in less number of gradients as compared to MAML.

$$\phi \rightarrow \phi + \epsilon(\phi^{'} - \phi) \tag{5}$$

3.4. Graph Neural Networks

In limited data learning, just learning the pattern effectively from a single data instance is not sufficient. We also need to learn the semantic relationships between the data instances itself (similar to notion of the learning the similarity). It has been observed in the literature (Zhou et al., 2020) that graph based learning are quite good in learning these contextual relations between several data instances itself. In order to capture this gap, we have built a Graphical Neural Network based model which will create an efficient embedding for each of the images and learn the relationships between them based on connections between them.

In our case considering graph structure, each images are nodes and are represented by a node embeddings based on representation policy and edges and its confounding variable edge weights are decided based on a connection policy, Then, GNN is applied to learn effectively the relations between them.5. As a representation policy we have used second last layer of GNN model and for connection policy we have used Euclidean distance threshold based pruning (Costa et al., 2021). We calculated the euclidean distance between each of the nodes using its node embeddings(6) and then connected the nodes with an undirected edge if the distance between them is lesser than that of average euclidean distance between nodes(7) in the entire dataset. We have also assigned weights to each of the edges as inverse of the euclidean distance between the two nodes(8) and finally each of the edges from a node are normalised as well(9).

After preparing our graph structure represented by nodes, nodes features, edges and edges variables, we have used two layers of GCN networks to further update the node embeddings based on the neighbouring nodes so as to understand the contextual relations between different images. We have developed a custom keras layer implementing our graph convolution layer based on the the Design space for GNN (You et al., 2020)

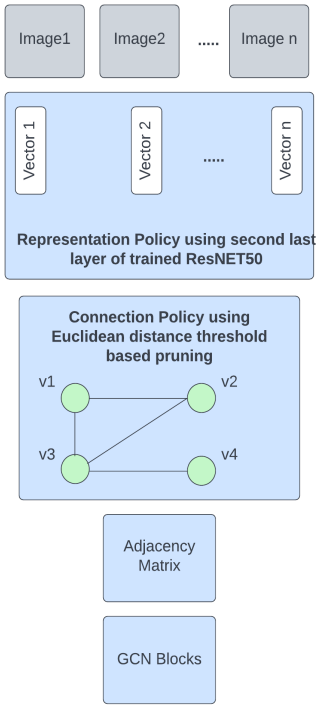

Figure 5. Graph based Learning

$$d_{i,j} = \sqrt{(\mathbf{v_i} - \mathbf{v_j})^2} \tag{6}$$

$$e_{i,j} = \begin{cases} 1 & d_{i,j} <= d_{avg}, \\ 0 & d_{i,j} > d_{avg} \end{cases} \tag{7}$$

$$w_{i,j} = \frac{1}{d_{i,j}} \tag{8}$$

$$w^N_{i,j} = \frac{w_{i,j}}{\Sigma_j w_{i,j}} \tag{9}$$

4. Experimental Settings

4.1. Baseline

For baseline and with added augmentations models, we have adopted a general policy of image pre-processing as instructed by the VIPriors challenge[2]. We have resized the input image to (256,256) and then did random cropping for training and validation to resize the image further to (224,224). Following this, we scaled each image pixel by factoring it from 255 and then finally normalized it with respect to Imagenet weights. We used stochastic gradient descent with learning rate scheduler of decreasing the value by a factor of 10 after 30 epochs starting from 0.1.

4.2. Similarity Learning

For similarity learning models, we have used the vector representation of ResNet-50 Baseline Models as a tabular dataset to speed up the training process. We have used a simple two layer MLP as embedding network which is then finally again trained a single layer perceptron and with the same compiler settings as that of baseline.

4.3. Meta Learning

In this model, we took ResNet-50 (baseline) representation of every image and use this tabular data as input for 5-way 5-shot classification i.e., we have taken 5 samples from each randomly sampled 5 classes and then this is treated as one classification task. We did 2000 such tasks, i.e., the maximum iterations were set to 2,000.

4.4. Graphical Neural Network

Even though we are dealing with limited data settings i.e, 50 images per class, still we have 1,000 class and thus total images are 50,000. The limited computation resource constrained us to create the graph of these many images. So as to simulate this model and demonstrate its effectiveness, we have randomly sampled 10 out of 1,000 classes and created the baseline for these 10 classes and then compared it with the GNN framework to see if it can gives us any improvement.We have used trained ResNet-50 baseline model which gives 2048 features for each images. To further observe the effectiveness of GNN, we clipped the first 100 features from the 2048 feature representations. We assume it gives the same effect as the dropout except it is always static to zero out everything except the first 100 features.

[2]https://github.com/VIPriors/vipriors-challenges-toolkit/tree/master/image-classification

5. Results and Analysis

5.1. Performance Evaluation

Model	Strategy	Validation	Test
BaseLine		28.79	26.95
Augmentation Techniques	CutMix	29.12	27.41
	MixUP	29.83	**28.08**
Similarity Learning	Siamese contrastive	38.16	22.28
	Siamese triplet	30.59	19.83
	Cosine similarity	11.97	10.53
	Supervised contrastive	23.30	16.08
Meta Learning	Reptile	**55.39**	27.23

Table 1. Performance Evaluation: Accuracy (in %) scores for validation and test set across different techniques and models.

Augmentation Techniques: We compare performance of all our experiments with the baseline. As shown in the table 1, we observe that MixUp augmentation technique performs with the highest accuracy score on the test set. Even though Yun et al. (2019a) claims that CutMix works better than MixUp, for our dataset, we find that MixUp performs slightly better than CutMix. We believe it is because of the efficiency of MixUp in domain agnostic applications. MixUp augments data by overlapping pixels of two images such that information at each pixels contains features from both the images in different proportions. While, CutMix augmentation completely removes a few pixels of the original image and replace it with the features of another image. We believe as ImageNet is multi-domain dataset, MixUp helps preserving information from both the images, hence helps in better learning.

Similarity-based Learning: On comparing results, we find that Siamese network with contrastive loss has the highest validation and test accuracy among all the similarity-based learning approaches we tried. Similarity-based learning approach is largely influenced by the pair mining techniques used. Siamese learning with contrastive and triplet loss comes under pair-mining techniques, where we require the mining techniques. We have observed that even though validation performance increased in both of them, test accuracy didn't improve. Siamese networks works very well in limited data setting. But in our case, the number of images per class is less than number of classes itself, and thus pair creation strategy that we have used is not able to explore all pairs at the time of training and thus doesn't generalises well on test data. Cosine similarity is not able to achieve the level of performance as mentioned in the (Barz & Denzler, 2020). Our dataset is very wide in terms of domain and comparison of features magnitude is required along with feature directions. This aspect is not examined in original paper as all of their datasets are very domain specific.

Meta Learning: We have tried REPTILE as one of the first order MAML technique. It can be observed that there is a high increase in the performance of validation set and it also performs slightly better on test set as well. The

reason of its effectiveness could be attributed to its ability to quickly learn the new tasks in limited data setting. It's basically an effective weight initialisation strategy, so the learned weights of a simple two layer MLP from meta learning when combined with even a softmax layer directly to output the class probability on query set and retrained again on support set, it leads to higher accuracy. One of the observation that we could make from Table 1, is that validation accuracy is higher compared to the test set. This is happening because while training we usually pick the weights from best validation performance epoch and thus adding bias towards the validation dataset, moreover we have observed that Fig. 6 at the epoch of maximum validation accuracy generalisation gap also shoots up and thus indicating the over-fitting scenario.[3]

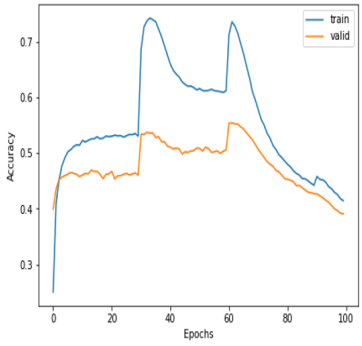

Figure 6. Learning curve for MLP classifier initialised from weights learned from Reptile.

5.2. Exploring GNNs

To demonstrate the effectiveness of GNN, we simulated a very low data setting as mentioned in section 4.4.[4] As shown in table 2, GNN gave us improvement of ~ 4% in both test as well as validation set. Randomly sampling 10 classes out of 100 have higher chances of selecting clearly distinguishable classes. This makes it difficult to claim the effectiveness of this method on the complete dataset. However, with our experiment we intuitively suggest usefulness of GNN in limited data setting as it helps learning the semantic relationship between different data points. To be able to provide stronger claims, further evaluation on the entire dataset with less class variance is required.[5]

Model	Validation	Test
Baseline	90.00	80.20
GNN	**94.00**	**85.40**

Table 2. Comparing GNN results with baseline, trained and test on the same dataset.

[3]One of our future works.

[4]Please note that these settings are different from the data settings of the original experiments.

[5]We are currently limited by our computational resource.

6. Conclusions

We explored multiple deep learning techniques including data augmentation, similarity-based learning, and meta-learning on the image classification track of VIPrior challenge. We found that straightforward data augmentation technique - Mixup, outperformed the baseline by 1.13%, while other techniques fell short on achieving closer performance. With REPTILE (meta-learning) we outperform the baseline on the validation set and slightly better in test set as well which could be attributed to its efficient weight initialisation or learning technique and thus speeding up the learning in limited data setting. Among the similarity-based techniques, we found that Siamese learning with contrastive loss outperformed the baseline on validation set but achieved lower performance on the test set. We believe that this is due to the large-number of classes in the dataset that influences the pair-mining techniques, consequently the training of the model. Additionally, we trained GNN model in a simulated low data setting (less data than provided) to study usefulness of GNN model in capturing data semantics. We found that there is an increase in performance in both validation and test set in the simulated limited data setting.

7. Future Works

We acknowledge the short-comings of a few techniques we addressed above, and in this section we discuss how we plan to improve on them. For Siamese-based similarity learning, we want to explore different data mining strategies like hard and soft pair/triplet mining to have more control on the mining strategies used, so as to better generalise on the test data. For the Meta Learning techniques we observed high gap between test and validation accuracy. To address this we wish explore different regularisation techniques like adding more dropouts, l1-l2 Normalisation or better compiling technique Gradient Centralisation (Yong et al., 2020). With GNN, given the required computational resources, we wish to apply it on the entire dataset and study its ability to generalization.

References

Barz, Björn and Denzler, Joachim. Deep learning on small datasets without pre-training using cosine loss. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1360–1369, 2020.

Barz, Björn and Denzler, Joachim. Deep learning on small datasets without pre-training using cosine loss. 2019. doi: 10.48550/ARXIV.1901.09054. URL https://arxiv.org/abs/1901.09054.

Bengio, Yoshua. Deep learning of representations for unsupervised and transfer learning. In Guyon, Isabelle, Dror, Gideon, Lemaire, Vincent, Taylor, Graham, and Silver, Daniel (eds.), *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pp. 17–36,

Bellevue, Washington, USA, 02 Jul 2012. PMLR. URL https://proceedings.mlr.press/v27/bengio12a.html.

Bruintjes, Robert-Jan, Lengyel, Attila, Baptista-Rios, Marcos, Kayhan, Osman Semih, and van Gemert, Jan. Vipriors 1: Visual inductive priors for data-efficient deep learning challenges. *pp. 511–520. Springer (2020)*.

Costa, Felipe F, Saito, Priscila TM, and Bugatti, Pedro Henrique. Video action classification through graph convolutional networks. In *VISIGRAPP (4: VISAPP)*, pp. 490–497, 2021.

Cubuk, Ekin D., Zoph, Barret, Mane, Dandelion, Vasudevan, Vijay, and Le, Quoc V. Autoaugment: Learning augmentation policies from data, 2018. URL https://arxiv.org/abs/1805.09501.

Cubuk, Ekin D., Zoph, Barret, Shlens, Jonathon, and Le, Quoc V. Randaugment: Practical automated data augmentation with a reduced search space, 2019. URL https://arxiv.org/abs/1909.13719.

Finn, Chelsea, Abbeel, P., and Levine, Sergey. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Han, Dongyoon, Yun, Sangdoo, Heo, Byeongho, and Yoo, YoungJoon. Rethinking channel dimensions for efficient model design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 732–741, June 2021.

Hinton, Geoffrey E., Vinyals, Oriol, and Dean, Jeffrey. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Khosla, Prannay, Teterwak, Piotr, Wang, Chen, Sarna, Aaron, Tian, Yonglong, Isola, Phillip, Maschinot, Aaron, Liu, Ce, and Krishnan, Dilip. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Kim, Byeongjo, Kim, Chanran, Lee, Jaehoon, Song, Jein, and Park, Gyoungsoo. Data-efficient deep learning method for image classification using data augmentation, focal cosine loss, and ensemble, 2020. URL https://arxiv.org/abs/2007.07805.

LeCun, Yann, Bengio, Y., and Hinton, Geoffrey. Deep learning. *Nature*, 521:436–44, 05 2015. doi: 10.1038/nature14539.

Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, and Dollár, Piotr. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

Luo, Z., Li G. Zhang Z. A technical report for vipriors image classification challenge (2020).

Luo, Yihao, Cao, Xiang, Zhang, Juntao, Cheng, Peng, Wang, Tianjiang, and Feng, Qi. Dynamic multi-scale loss optimization for object detection, 2021.

Nichol, Alex, Achiam, Joshua, and Schulman, John. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. URL http://arxiv.org/abs/1803.02999.

Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Pengfei Sun, Xuan Jin, Xin He Huiming Zhang Yuan He Hui Xue. A technical report for vipriors image classification challenge (2020).

Ridnik, Tal, Lawen, Hussam, Noy, Asaf, Ben Baruch, Emanuel, Sharir, Gilad, and Friedman, Itamar. Tresnet: High performance gpu-dedicated architecture. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1400–1409, January 2021.

Romero, Adriana, Ballas, Nicolas, Kahou, Samira Ebrahimi, Chassang, Antoine, Gatta, Carlo, and Bengio, Yoshua. Fitnets: Hints for thin deep nets, 2014. URL https://arxiv.org/abs/1412.6550.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. Imagenet large scale visual recognition challenge, 2014. URL https://arxiv.org/abs/1409.0575.

Sun, P., Jin X. Su W. He Y. Xue H. Lu Q. A visual inductive priors framework for data-efficient image classification. in: European conference on computer vision workshops. *CoRR*, abs/2103.03768, 2021. URL https://arxiv.org/abs/2103.03768.

Suprapto and Polela, Joseph A. The influence of loss function usage at siamese network in measuring text similarity. *International Journal of Advanced Computer Science and Applications*, 11(12), 2020. doi: 10.14569/IJACSA.2020.0111290. URL http://dx.doi.org/10.14569/IJACSA.2020.0111290.

Szegedy, Christian, Ioffe, Sergey, Vanhoucke, Vincent, and Alemi, Alexander A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

Tan, Mingxing and Le, Quoc. EfficientNet: Rethinking model scaling for convolutional neural networks. In Chaudhuri, Kamalika and Salakhutdinov, Ruslan (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/tan19a.html.

Tan Wang, Wanqi Yin, Jiaxin Qi Jin Liu Jayashree Karlekar Hanwang Zhang. A technical report for vipriors image classification challenge (2020).

Utkin, Lev, Kovalev, Maxim, and Kasimov, Ernest. An explanation method for siamese neural networks. In *Proceedings of International Scientific Conference on Telecommunications, Computing and Control*, pp. 219–230. Springer, 2021.

Vanschoren, Joaquin. Meta-learning: A survey, 10 2018.

W Zi, S Prince. few-shot learning and meta-learning i, 10 2019.

Wan, Alvin, Dunlap, Lisa, Ho, Daniel, Yin, Jihan, Lee, Scott, Jin, Henry, Petryk, Suzanne, Bargal, Sarah Adel, and Gonzalez, Joseph E. URL https://arxiv.org/abs/2004.00221.

Wang, Feng and Liu, Huaping. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

Wen, Yandong, Zhang, Kaipeng, Li, Zhifeng, and Qiao, Yu. A discriminative feature learning approach for deep face recognition, 2016.

Yong, Hongwei, Huang, Jianqiang, Hua, Xiansheng, and Zhang, Lei. Gradient centralization: A new optimization technique for deep neural networks, 2020. URL https://arxiv.org/abs/2004.01461.

You, Jiaxuan, Ying, Rex, and Leskovec, Jure. Design space for graph neural networks. *CoRR*, abs/2011.08843, 2020. URL https://arxiv.org/abs/2011.08843.

Yun, Sangdoo, Han, Dongyoon, Oh, Seong Joon, Chun, Sanghyuk, Choe, Junsuk, and Yoo, Youngjoon. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019a.

Yun, Sangdoo, Han, Dongyoon, Oh, Seong Joon, Chun, Sanghyuk, Choe, Junsuk, and Yoo, Youngjoon. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019b.

Zagoruyko, Sergey and Komodakis, Nikos. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, 2016. URL https://arxiv.org/abs/1612.03928.

Zhang, Hang, Wu, Chongruo, Zhang, Zhongyue, Zhu, Yi, Zhang, Zhi, Lin, Haibin, Sun, Yue, He, Tong, Mueller, Jonas, Manmatha, R., Li, Mu, and Smola, Alexander. Resnest: Split-attention networks, 04 2020.

Zhang, Hongyi, Cisse, Moustapha, Dauphin, Yann N., and Lopez-Paz, David. mixup: Beyond empirical risk minimization, 2017a. URL https://arxiv.org/abs/1710.09412.

Zhang, Hongyi, Cisse, Moustapha, Dauphin, Yann N, and Lopez-Paz, David. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017b.

Zhou, Jie, Cui, Ganqu, Hu, Shengding, Zhang, Zhengyan, Yang, Cheng, Liu, Zhiyuan, Wang, Lifeng, Li, Changcheng, and Sun, Maosong. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000012.