

MixMatch Domain Adaptation: Prize-winning solution for both tracks of VisDA 2019 challenge

Danila Rukhovich

d.rukhovich@samsung.com

Danil Galeev

d.galeev@samsung.com

Abstract

We present a domain adaptation (DA) system that can be used in multi-source and semi-supervised settings. Using the proposed method we achieved **2nd** place on multi-source track and **3rd** place on semi-supervised track of the VisDA 2019 challenge¹.

1. Introduction

Unsupervised domain adaptation aims to generalize a model learned from a source domain with rich annotated data to a new target domain without any labeled data. To speed up research progress in this area the DomainNet [3] dataset was released. This dataset contains around 0.6 million images of 345 classes and 6 domains: *real*, *infograph*, *quickdraw*, *sketch*, *clipart* and *painting*. The VisDA 2019 challenge aims to test new domain adaptation algorithms on this dataset. For multi-source track the task is to train model on image from 4 annotated domains and 2 unlabeled domains to maximize accuracy on these 2 target domains. For semi-supervised track a very few (3 per class) labeled images from 2 target domains are available for training and only *real* is used as a source domain.

2. Proposed method

In short, our method can be characterized as MixMatch with EfficientNet backbone. In this section we present a brief description of these 2 architectures and our contribution to their application to the contest tasks. General scheme of our approach is shown in Figure 1.

2.1. MixMatch

MixMatch [1] is a state-of-the-art method in semi-supervised learning. Its main idea is the combining of currently dominating approaches in this field like multiple augmentations, label guessing, label distribution sharpening and MixUp [4]. The authors conducted experiments show-

ing the effect of each of these components on the final classification accuracy. Borrowing this insights we show how this method can be applied to domain adaptation tasks.

2.2. EfficientNet

Although MixMatch and M3SDA [3] use ResNet architectures (Wide Resnet-28 and Resnet-101) as a backbone for their methods, we decided to use state-of-the-art ImageNet classification architecture EfficientNet [2]. This model is a result of neural architecture search with carefully balancing of network depth, width and resolution. It is also shown that better resulting models from image classification and transfer learning have even less number of parameters. One more advantage of using this network is the set of 8 pretrained models: EfficientNet-b0, ..., EfficientNet-b7 with increasing number of parameters and overall accuracy. In our experiments first five models from this list didn't generalized well, and the last one EfficientNet-b7 was too heavy, which necessitated a reduction in the size of the batches and resulted a decrease in accuracy.

2.3. Overview

As the MixMatch is not a domain adaptation method, we propose a new scheme for constructing mini-batches during training. Originally, batch contains n labeled image, n unlabeled images and n same unlabeled images with different random augmentation. For our approach unlabeled part transforms to target domain with no changes. For multi-source domain adaptation we construct the labeled part of batch from $\frac{n}{k}$ images from each of k source domains. For semi-supervised domain adaptation the labeled part of batch consists of $\frac{4n}{5}$ source images and $\frac{n}{5}$ images from labeled part of target domain.

We also propose a training process modification to improve its stability. During one MixMatch training step the backbone model is called 5 times: 2 times for target domain batches and 3 times for mixed up batches. The distributions in these 3 batches differ significantly, because one is dominated by source images and the other two by target images. This, in turn, causes instability when updating the batch normalization layer statistics. We form each batch of $\frac{1}{3}$ source

¹<http://ai.bu.edu/visda-2019/>

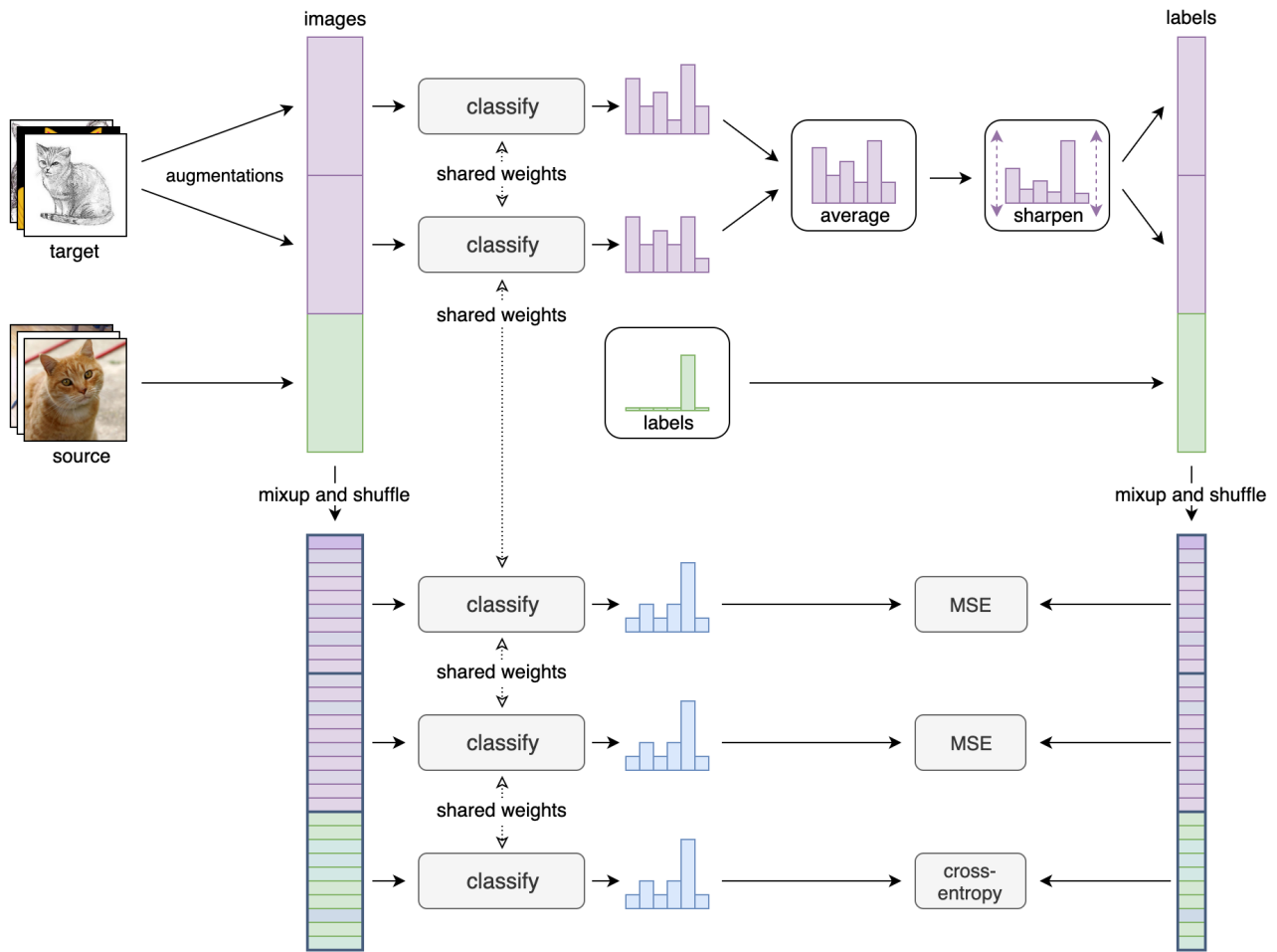


Figure 1. Scheme of the proposed method.

and $\frac{2}{3}$ target images to approximate the statistics between the batches entering the network.

In addition to experimenting with the MixMatch model, we also tested our baseline. This approach only trains EfficientNet on all labeled data. Comparison of this 2 models is presented in section 3.2.

We use the same data augmentation strategy during training and testing time:

1. resize to 256×256 pixels,
2. random horizontal flip,
3. random crop of size 224×224 pixels.

2.4. Model ensembling

It is widely known that the ensemble averaging of neural networks trained independently leads to the improvement of test accuracy. In this work we trained models with different EfficientNet backbones and different weight of loss balancing (cross-entropy and mean squared). We used equal

average of predictions from these models to make final prediction. In our experiments this technique gave an increase in accuracy for both tasks of the challenge.

3. Experiments

3.1. Training details

We implemented our MixMatch-based and baseline models in Tensorflow 2.0 from scratch. We used open source EfficientNet pre-trained on ImageNet models². For all experiments without ensembling we used same hyper-parameters. We trained the network with Adam optimizer with 0.0001 learning rate and batch size of 15 for 100 epochs (epoch is 1000 batches; 10 epochs is enough for baseline models). MixMatch parameters (except of loss weight) are set to their default values from the original paper, in particular, beta distribution parameter of 0.75 and

²<https://github.com/qubvel/efficientnet>

label sharpening power of 0.5. Loss weight parameter is a multiplier for mean squared error, we set its default value to 333.0. For all experiments we used 7 test time augmentations.

We trained our models on 8 Nvidia Tesla P40 GPUs with 24 Gb memory each, which allows us to use $15 \times 3 \times 8$ images per one optimizer update step. We noticed that smaller batch size leads to decreasing of target accuracy.

3.2. Validation

During the validation phase of the competition *sketch* was the target domain, for multi-source (ms) track *real*, *quickdraw* and *infograph* were the source domains and for semi-supervised (ss) track only *real* was the source domain.

To demonstrate the benefits of domain adaptation, we compare the MixMatch model with baseline (not using DA). As can be seen from Table 1, the growth of target accuracy on both tracks is about 10%. For semi-supervised track we also show the benefit of using the labeled part of target domain even for baseline model. Here we use MixMatch model with EfficientNet-b5 backbone.

track	model	accuracy
ms	baseline	0.519
ms	MixMatch	0.619
ss	baseline w/o labeled target	0.457
ss	baseline	0.520
ss	MixMatch	0.639

Table 1. Evaluation of different models for validation phase.

3.3. Testing

During the testing phase of the competition *clipart* and *painting* were used as the target domain, for multi-source track *real*, *quickdraw*, *infograph* and *sketch* were the source domains and for semi-supervised track only *real* was the source domain. We trained same models for 2 target domains and then concatenated predictions for final submission.

Our results are shown in table 2. For both multi-source and semi-supervised tracks we trained 3 models with 2 different backbones and loss weights. The accuracy of all models is almost equal, and model ensembling gives 1% profit.

With these results, we achieved the prize-winning places of the VisDA 2019 challenge. Top 3 results for both tracks are shown in Table 3 (our team name is *denemmy*).

References

[1] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. Mixmatch: A holistic approach to

track	EfficientNet	loss weight	accuracy
ms	b5	333.0	0.699
ms	b5	1000.0	0.689
ms	b6	333.0	0.695
ms	ensemble		0.716
ss	b5	333.0	0.704
ss	b5	1000.0	0.695
ss	b6	333.0	0.703
ss	ensemble		0.713

Table 2. Evaluation of MixMatch model with different parameters for test phase.

track	team	accuracy
ms	Yingwei.Pan	0.760
ms	denemmy	0.716
ms	numpee	0.696
ss	lunit	0.720
ss	Yingwei.Pan	0.714
ss	denemmy	0.713

Table 3. Top 3 places on final leaderboard.

semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019. 1

[2] T. Mingxing and L. Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 1

[3] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. *arXiv preprint arXiv:1812.01754*, 2018. 1

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1