Universal Domain Adaptation and Open Set Classification with focus on Visual Domain Adaptation Challenge 2021

Jiawei Liu
Du Xiaoman Financial
Beijing, China
liujiawei@duxiaoman.com

Qing Yang
Du Xiaoman Financial
Beijing, China
yangqing@duxiaoman.com

Abstract

We presented a universal domain adaptation system based on pseudo label for domain adaptation and open set classification. We used OVANet architecture for in-class classification and out-of-class confidence prediction, and then trained the models for multiple times with unlabeled data with pseudo-labels for higher performance. In addition, we trained multiple different models and implemented model ensemble by voting on the predicted results of these models. Finally, our method ranks 4th place in the NeurIPS 2021 Visual Domain Adaptation Challenge(VisDA-2021).

1. Introduction

Progress in machine learning is typically measured by training and testing a model on the same distribution of data, i.e., the same domain. However, in real-world applications, models often encounter out-of-distribution data, such as novel camera viewpoints, backgrounds or image quality.

Universal domain adaptation (UDA) aims to generalize a model learned from a source domain with rich labeled data to a new target domain without any labeled data. In addition, the target data may also have missing and/or novel classes, this introduces open set classification on the basis of UDA.

The NeurIPS 2021 Visual Domain Adaptation Challenge(VisDA-2021) aims to test the algorithm's ability in domain adaptation and open set classification[1]. VisDA-2021 provides training sets, development sets and test sets to develop models. Training set (source data) is images and object labels from the standard ImageNet 1000-class training dataset. Development set (target data) is images and labels sampled from ImageNet-C (orruptions), ImageNet-R (enditions) and ObjectNet. The dev set contains some (but not necessarily all) source classes and some novel classes. Test set is similar to the development set but with a different input distribution and category composition. No labels are released for the test set.

2. Proposed method

In short, our method can be represented as the introduction of semi-supervised learning method on the basis of OVANet [1]. We tried to use pseudo label method [2], consistent regular method [3], mean teacher method [4] and MixMatch [5] and finally chose the pseudo label method. We used different levels of EfficientNet [6] as backbone, and finally used voting method for model ensemble.

2.1 Model

[1] http://ai.bu.edu/visda-2021/

We built the model based on the OVANet architecture. The closed set classifier C1 is used to predict the classification within 1000 classes, and the open set classifier C2 is used to generate the anomaly score outside 1000 classes. For source data, the losses of both closed and open set classifiers are calculated. For data with pseudo label, the loss of closed set classifier and the loss of information entropy generated by open set classifier are calculated.

We used EfficientNet-B6, EfficientNet-B5 and EfficientNet-B4 (advprop version, pre-trained model based on adversarial training) pre-trained on ImageNet as backbone (G in OVANet), corresponding to training three models respectively[2]. The output dimension of the three EfficientNet is 500, so that the number of model parameters of C1 is 0.5M and that of C2 is 1M. Numbers of model parameters are:
the model with EfficientNet-B6 as G:
41.8882+0.5+1=43.3820M
the model with EfficientNet-B5 as G:
29.3653+0.5+1=30.8653M
the model with EfficientNet-B4 as G:
18.4451+0.5+1=19.9451M
The total number of parameters is:
43.3820+30.8653+19.9451=94.1924M<100M

2.2 Data Augmentation

The images were scaled, flipped horizontally with probability of 0.5, randomly cropped and normalized, without using other data augmentation techniques. For the different EfficientNet options, the scaling and random clipping sizes are different, as shown in Table 1.

Table 1. The corresponding scale size and random crop size of different models

Model	Scale size	Random crop size
the model with EfficientNet-B6 as G	560	528
the model with EfficientNet-B5 as G	488	456
the model with EfficientNet-B4 as G	412	380

2.3 Training

1) Pre-training

Models are trained by source data and target data to obtain the pre-training model. For source data, the losses of both closed and open set classifiers are calculated. Information entropy loss is calculated for target data. Figure 1 shows an overview of pre-training process.

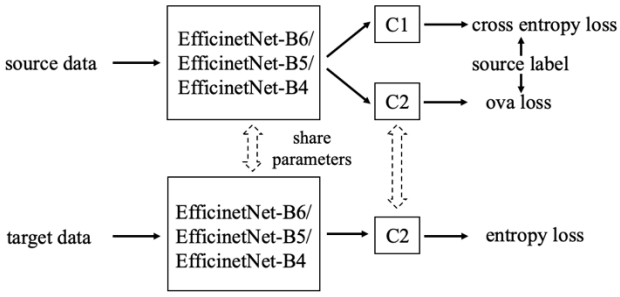

Figure 1. An overview of pre-training process

2) Stage 1 Training

a. Pseudo-label Prediction: The models obtained by pre-training are used to test the target data to get the category and anomaly score, and the category is used as the pseudo label of the target data to participate in the subsequent training.

b. Different from pre-training, pseudo-label training introduces classification loss of target data (Only closed set loss is included. We also tried to introduce the open set loss of target data, but experiments show that the accuracy is not as good as without.).

c. Repeat a, b twice.

3) Stage 2 Training

a. Pseudo-label Prediction: The models obtained by stage 1 training are used to test the test data to get the category and anomaly score, and the category is used as the pseudo label of the test data to participate in the subsequent training.

b. Training: Introduced classification loss of test data (Only closed set loss is included).

[2] https://github.com/lukemelas/EfficientNet-PyTorch

c. Repeat a, b four times.

Figure 2 shows an overview of stage 1/2 training process.

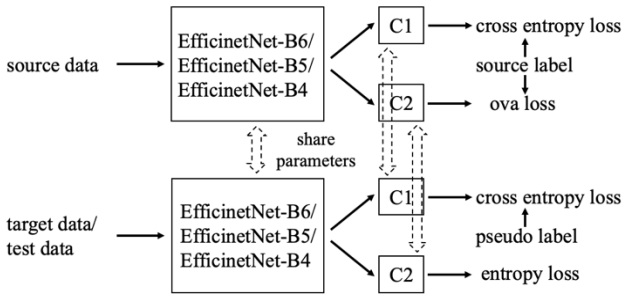

Figure 2. An overview of stage 1/2 training process

2.4 Model Ensemble

After training, we obtained the corresponding prediction results of the three models. We realized model ensemble through voting, and the algorithm of model ensemble is shown in Algorithm 1.

Algorithm 1 Model Ensemble

input:

The prediction of the model with EfficientNet-B6 as G which category and anomaly score are pred1 and score1.

The prediction of the model with EfficientNet-B5 as G which category and anomaly score are pred2 and score2.

The prediction of the model with EfficientNet-B4 as G which category and anomaly score are pred3 and score3.

output: The final prediction file which category and anomaly score are pred and score.

```
if pred1 != pred2 and pred2 != pred3:
    pred = pred1, score = score1
elif pred1 == pred2 or pred1 == pred3:
    pred = pred1, score = score1
elif pred2 == pred3 and pred1 != pred2:
    pred = pred2, score = score2
```

3. Experiments

3.1 Training details

We implemented our models in PyTorch. For all experiments we used same hyperparameters. We trained the networks with SGD optimizer with momentum of 0.9 and weight decay of 0.0005 and nesterov is True. The learning rate of G and C1/C2 are 0.001 and 0.01, batch size is 8 and step is 20000, the weight factor of entropy loss is 0.05. We trained our models on 2 NVIDIA V100 GPUs with 16GB memory each.

For the training phase involving pseudo-label data, we set a weight of pseudo-label data loss. Due to poor model performance at the early stage of training, this weight is relatively small and increases gradually during training, and stabilizes at a value at the later stage of training, as shown in the Eq.1, where alpha is the weight of pseudo-label data loss, t1=200, t2=1000, af=0.6.

$$\text{alpha} = \begin{cases} 0, & step < t1 \\ \dfrac{step - t1}{t2 - t1} \times af, & t1 \le step \le t2 \\ af, & step > t2 \end{cases} \quad (1)$$

3.2 Results

1) Pre-training

The prediction accuracy of the development set on three pre-training models is shown in Table 2.

Table 2. The prediction accuracy of the development set on three pre-training models.

Model	Accuracy	AUROC
the model with EfficientNet-B6 as G	48.61%	65.90%
the model with EfficientNet-B5 as G	49.65%	66.15%
the model with EfficientNet-B4 as G	46.58%	67.34%

2) Stage 1 Training

The prediction accuracy of the development set on the stage 1 training models is shown in Table 3.

Table 3. The prediction accuracy of the development set on the stage 1 training models.

Model	Accuracy	AUROC
the model with EfficientNet-B6 as G	58.97%	68.60%
the model with EfficientNet-B5 as G	57.26%	68.51%
the model with EfficientNet-B4 as G	52.11%	64.16%

3) Stage 2 Training

Since the real label of the test set is not visible in this stage of training, it is impossible to know the respective prediction accuracy of the three models. Table 4 shows the prediction accuracy after model ensemble. By contrast, the prediction accuracy of only source data is obtained by training the model with only source data once, which also shows in Table 4. It can be seen that our method greatly improves the prediction results compared with only the source data, indicating that the performance of the model depends not only on the basic capability of the model but also on the domain adaptability.

Table 4. The prediction accuracy of test set on domain adaptation model and source-only model (final results on leaderboard)

Method	Accuracy	AUROC
Adapt pred	48.60%	68.29%
Source only pred	25.70%	62.43%

4. Conclusion

In this abstract, we presented our universal domain adaptation method based on semi-supervised learning. By using this method, the prediction accuracy of our model is greatly improved compared with the model trained with only source data.

References

[1]. Saito K, Saenko K. OVANet: One-vs-All Network for Universal Domain Adaptation[J]. 2021.

[2]. Lee D H. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. 2013.

[3]. Xie Q, Dai Z, Hovy E, et al. Unsupervised Data Augmentation for Consistency Training[J]. 2019.

[4]. Tarvainen A, Valpola H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results[J]. 2017.

[5]. Berthelot D, Carlini N, Goodfellow I, et al. Mixmatch: A holistic approach to semi-supervised learning[J]. arXiv preprint arXiv:1905.02249, 2019.

[6]. Tan M, Le Q V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks[J]. 2019.