Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 10, Number 1, pages 1–3, January 2021

Transfer Learning Algorithm for Object Detection

1st Yuta Suzuki
Kanazawa University
Kanazawa, Japan
ysuzuki@csl.ec.t.kanazawa-u.ac.jp

2nd Daiki Kuyoshi
Kanazawa University
Kanazawa, Japan
dkuyoshi@csl.ec.t.kanazawa-u.ac.jp

3rd Satoshi Yamane
Kanazawa University
Kanazawa, Japan
syamane@is.t.kanazawa-u.ac.jp

***Abstract*—This study is related to transfer learning in Faster-RCNN, which is a representative model for object detection tasks. Image recognition includes image classification, object detection and image segmentation task. Transfer learning is especially important for the object detection task and the image segmentation task because of the high cost of generating training data. In this study, we use an algorithm to calculate the difference between tasks by focusing on the amount of parameter updates. We then applied the algorithm to an object detection task and aimed to make it useful for transfer learning.**

I. Introduction

In recent years, rapid development of machine learning techniques has a great impact on the image recognition field. Among them, the object detection task is used for automatic driving and other applications, and is an active task. [1] However, it is difficult to prepare enough training data for the object detection task. In order to solve the problem, the method of transfer learning is very important. Transfer learning is a technique to make the features learned in other domains useful for learning in other domains, and is effective for learning small training data sets. However, the pre-trained model must have been trained on a large dataset. [2] For this purpose, it is important to make transfer learning effective using the models pre-trained by ImageNet (14 million images) [3] and COCO2017 (330,000 images).

II. Related Works

A. *fine tuning*

Fine tuning is a commonly used transfer learning technique in CNN. Fine tuning freezes the parameters in the part of learning generic features. Then, the part that learns specific features that may differ from task to task updates the parameters. This prevents overfitting to a target task with a small number of data. In general, a layer close to the input of the CNN prohibits parameter updates in that layer as it learns generic features of the image, and allows parameter updates in the layer close to the output. However, the method of determining the layer that prohibits updating parameters is only empirical and is practically a hyper-parameter [4].

B. *Transfer Learning Algorithm in U-Net*

The study of [5] considered that when initialized with the parameters of the pre-trained model, they considered that the amount of parameter updates due to learning of the target task represents the difference in the features to be learned. In this study, They compute the difference between the parameters wi learned in ImageNet and wi' learned in the target task in layer i, and propose an algorithm to visualize the difference in training features between the two layers. The cosine similarity, which is used in statistics, is employed as a measure of parameter differences. As a result, they found that there is a significant difference between the training features of the ImageNet and the image segmentation task in the part that includes location information, and they applied fine tuning to U-Net by retraining this part.

III. Proposed Method

In this study, we apply the transfer learning algorithm III-B based on the work of [5] to a representative object detection model, Faster-RCNN [6]. We then visualize the differences between the training features at each layer in the object detection task and the pre-trained model. And the aim is to help select a pre-trained model and decide which layers to prohibit parameter updates.

A. *The Faster-RCNN model*

The model used was the Faster-RCNN as shown in Fig. 1 with a ResNet50 [8] consisting of 48 convolutional layers.

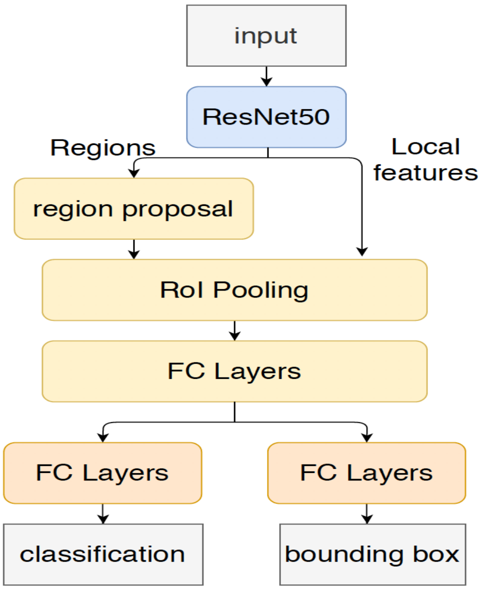

Fig. 1. The Faster-RCNN model we used

B. Algorithm for transfer learning

The explanation for Algorithm 1 is as follows. In the convolutional layer, the difference between the parameters wi learned in the pre-training task and wi' learned in the target task is calculated using cosine similarity. The algorithm proposed in the study of [5] included an operation to prohibit updating the parameters in the layer where the calculated difference exceeds the threshold t. However, that operation was omitted in this study because there was no clear way to set the threshold.

Algorithm 1

```
Ensure: $cos(\mathbf{w}', \mathbf{w})$
  // cos is Cosine similarity
  // t is the threshold value
  // W is a pretrained parameter
  // W' is a parameter trained in new task
  $\mathbf{w_0}, \mathbf{w_1}, ..., \mathbf{w_n} \Leftarrow \mathbf{W}$
  $\mathbf{w'_0}, \mathbf{w'_1}, ..., \mathbf{w'_n} \Leftarrow \mathbf{W'}$
  // Storage of parameters at each layer
  while $i \leq n$ do
    $l_i \Leftarrow |1.0 - cos(\mathbf{w'_i}, \mathbf{w_i})|$
    $i \Leftarrow i + 1$
  end while
```

IV. Experiments and Results

A. Methods

Kaggle's Global Wheat Detection [7] was used as an object detection task in the evaluation experiments. This is a task to detect wheat ears from wheat images, with more than 3,000 training data and about 1,000 test data. The model used was the Faster-RCNN as shown in Fig. 1 with a ResNet50 [8] consisting of 48 convolutional layers. We experimented on transfer learning with ImageNet and COCO2017 as a pre-training task in the ResNet50 part of the model used. Then, we apply Algorithm1 to the ResNet50 part and compare the results.

B. Results

Fig. 2 is the result of calculating the difference between the pre-learning task and the target task in each layer by the procedure of Algorithm 1. The horizontal axis is the index of the convolution layer of ResNet50 from the input side. The vertical axis is the difference between tasks calculated using cosine similarity. The orange line represents the difference between the parameters pre-trained by COCO2017 and those trained by Global Wheat Detection. The blue line represents the difference between the pre-trained parameters in ImageNet and those trained by Global Wheat Detection. Fig. 2 shows that the difference between the parameters between COCO2017 and Global Wheat Detection is relatively large from around the 20th layer to the output layer. On the contrary, the difference between the parameters between ImageNet and Global Wheat Detection is relatively small from around the 20th layer to the output layer.

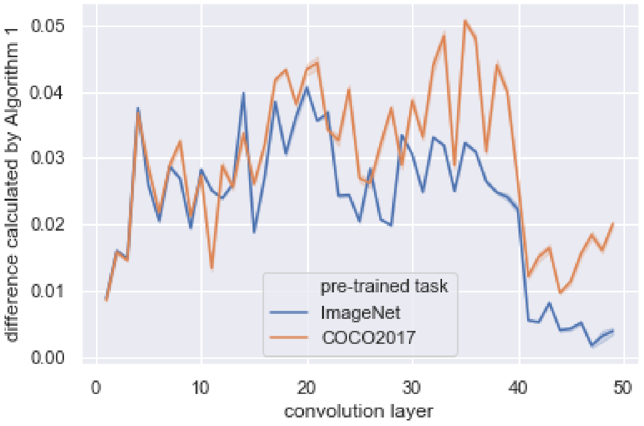

Fig. 2. Each Layer Difference

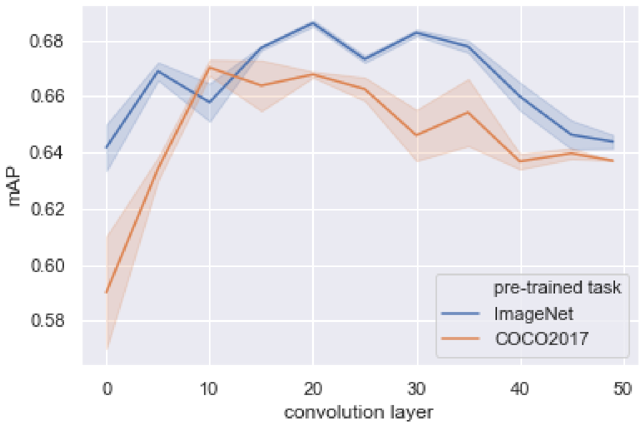

Fig. 3. Fine-Tuning Results

Fig. 3 is the result of applying fine tuning to the ResNet50 part of Fig. 1. The horizontal axis is the number of layers where the parameter update is prohibited from the input side, and the fine tuning is applied in increments of 5 layers. The vertical axis is the mAP (Mean Average Precision) which is an index to compare the accuracy of the object detection. The orange line is the result of pre-trained by COCO2017 and applying fine tuning. The blue line is the result of pre-trained by ImageNet and applying fine tuning. Fig. 3 shows that the model pre-trained by ImageNet has a higher mAP in all cases except for the case where the update of 10-layer parameters is prohibited. Therefore, we can say that the model pre-trained by ImageNet is more suitable for pre-training of Global Wheat Detection. From the results in Fig. 3, we can also read the relationship between the number of layers where parameter updates are prohibited and the mAP at that time. There was no significant difference between ImageNet and COCO2017 when parameter updates were prohibited for 10 to 15 layers, but when parameter updates were prohibited for more layers, the model pre-trained by ImageNet recorded a higher mAP.

From the results of Fig. 2 and Fig. 3, we can see that it is more effective to adopt a model with small differences as a pre-training model. It can also be seen that prohibiting the update of the weights of the layers with large differences reduces the performance.

V. Conclusion

The algorithm for calculating differences between tasks, which was applied to the segmentation task in study [5], was applied to the object detection task in this study. And we aimed to help transfer learning with pre-trained models at ImageNet and COCO2017. As a result, we are able to calculate the differences between tasks in the object detection task. And it could help in the selection of a pre-trained model and the decision on which layers to prohibit parameter updates.

References

[1] Li Liu et al., ”Deep Learning for Generic Object Detection: A Survey”, International Journal of Computer Vision volume (IJCV) 128, pages261-318, 2020.

[2] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, Samy Bengio, ”Transfusion: Understanding Transfer Learning for Medical Imaging”, Advances in Neural Information Processing Systems 32 (NIPS), 2019.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, ”ImageNet: A Large-Scale Hierarchical Image Database”, IEEE Computer Vision and Pattern Recognition (CVPR), 2009.

[4] Qiang Yang, Yu Zhang, Wenyuan Dai, Sinno Jialin Pan, Cambridge University Press, ”Transfer Learning”, pp. 54-55, 2020 in press.

[5] Yuta Suzuki, Satoshi Yamane, ”Transfer Learning Model for Image Segmentation by Integrating U-Net++ and SE Block”, IEEE 9th Global Conference on Consumer Electronics (GCCE), 2020.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, ”Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, arXiv:1506.01497v3, 2016.

[7] E. David, S. Madec, P. Sadeghi-Tehran, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, K. Nagasawa, M.A. Badhon, C. Pozniak, B. de Solan, A. Hund, S.C. Chapman, F. Baret, I. Stavness, W. Guo, ”Global Wheat Head Detection (GWHD) dataset: a large and diverse dataset of high resolution RGB labelled images to develop and benchmark wheat head detection methods”, arXiv:2005.02162v2, 2020.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, ”Deep Residual Learning for Image Recognition”, arXiv:1512.03385v1, 2015.