Pre-Training Transformers for Domain Adaptation

Burhan Ul Tayyab *[1] and Nicholas Chua[1]

[1]Shirley Robotics

Abstract

The Visual Domain Adaptation Challenge 2021 called for unsupervised domain adaptation methods that could improve the performance of models by transferring the knowledge obtained from source datasets to out-of-distribution target datasets. In this paper, we utilize BeiT [1] and demonstrate its capability of capturing key attributes from source datasets and apply it to target datasets in a semi-supervised manner. Our method was able to outperform current state-of-the-art (SoTA) techniques and was able to achieve 1st place on the ViSDA Domain Adaptation Challenge with ACC of 56.29% and AUROC of 69.79%.

1 Introduction

1.1 Visual Domain Adaptation

Traditional deep learning methods work really well in a constrained environment where the target dataset is close to the source dataset on which it is trained on. However, as demonstrated by [2], any shift in attributes (viewpoints, lightning conditions, orientations etc) and/or shift in label classes (where the target set varies/has new classes which aren't present in source dataset) would could cause the model to perform poorly and the accuracy to drop significantly. This could lead to a lot of problems in real-life scenarios if didn't taken into account. To solve this problem, we utilize BeiT [1], and demonstrate that it could self-learn various attributes by itself and can be adapted on new target datasets.

1.2 Related Work

Following AlexNet[3], convolutional neural networks (CNNs) have become standard for image classification tasks. Various models based on CNNs[4][5][6]

*Corresponding author: burhan@shirleyrobotics.com

have been introduced that achieve a significant increase in accuracy on various datasets[7]. However, these models fail to perform in conditions where there is a big input distribution shift between training and testing dataset and/or has label set variance[2]. Transformers such as BiT[8], ViT[9] have demonstrated significant improvement over CNNs in terms of accuracy at image classification tasks, however they require huge amounts of data to train. Meanwhile[10] utilizes the idea of using a single fixation of parse trees for image classification and attribute learning and[11] improves that by using dynamic routing and fixed vector representation for image classification, however both of the ideas don't work significantly well on large datasets because of its nonlinearity which results an increase in training complexity.

Domain Adaptation refers to fitting a model that has been trained on a particular source dataset on an out-of-source novel target distribution, which is not part of the training set. Closed-Set Domain Adaptation[12][13] methods, where the source and the target domain completely share the class of their samples, work extensively well and have low input distribution shifts, however they fail to work in open-set environments because of unknown target input samples. Self-supervised learning methods could be used to solve these issues by either distillation[14] or contrastive learning[15], however these methods have significant drawbacks[16][17].

2 Proposed Approach

In this section, we will show the proposed method in detail. Figure-1 gives an overview of the method utilized by us.

2.1 Model: BeiT

BERT[18], along with its Masked Language Modelling (MLM) module has performed wonders in the Natural Language Processing (NLP) domain. Inspired by BERT, we utilize BeiT-B[1] for performing universal domain adaptation. The input image is preprocessed and converted into patches, while is also simultaneously tokensized by DALL-E[19]. The patches are then masked randomly and fed to the BeiT-B Encoder which outputs hidden embeddings which are reconstructed by Masked Image Modelling Head by using input tokens. Since the idea is to just train the method on ImageNet-1k and test it on a related but un-constrained dataset with open world settings, we chose this method because it resembles masked language modelling in BERT. The whole system is pre-trained on ImageNet-1k, where the Masked Image Modelling module is able to reconstruct the corrupted patch via self-supervised self-attention and thus is able to recognize and separate the image without using any labels. After pretraining, a classification head is attached to the model where the pretrained model is fine-tuned to perform image classification.

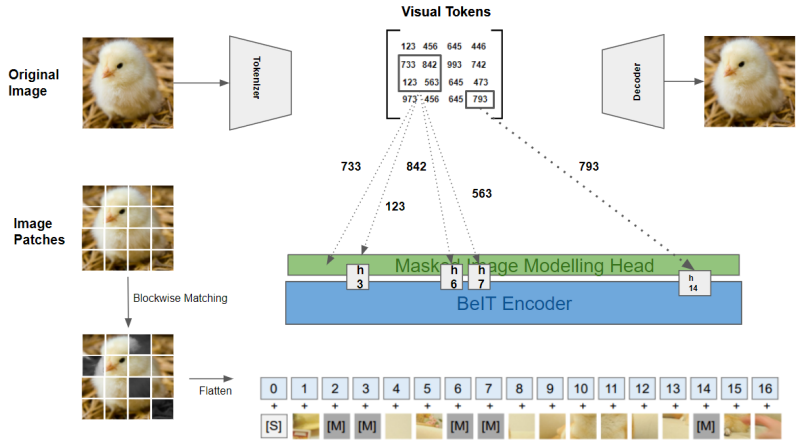

Figure 1: BeiT Architecture

2.2 Preprocessing

For pretraining, the dataset was resized into 224 x 224 and preprocessed with random cropping, color jittering, and horizontal flipping. The input image is then split into 14 x 14 images patches and masked via block-wise masking[1]. Simultaneously, the input image is also tokenized via [19] which would later be utilized for performing visual patch reconstruction by Masked Image Modelling head.

2.3 Methods

2.3.1 L-Layer ViT (BeiT)

We denote a dataset of images as $X = \{x_1, ..., x_N\}$ where N represents the total number of images and $\forall i \in \{1, ..., N\},\ x_i \in \mathbb{R}^{H \times W \times C}$. For image x_i, we will splice it into d number of image patches which is represented as $x_i^p \in \{x_{(i,1)}^p, ..., x_{(i,d)}^p\}$ where each patch is denoted by $x_{(i,k)}^p \in \mathbb{R}^{P \times P \times C}$, the number of patches are $d = \frac{HW}{P^2}$, and P^2 is the area of the patch. After slicing, the patches are masked randomly via blockwise masking algorithm [1]. The patches are then flattened to form a matrix, $v_i^p \in \mathbb{R}^{(P^2C) \times d}$. The embedding matrix $E \in \mathbb{R}^{D \times (P^2C)}$ and position embedding matrix $E_{pos} \in \mathbb{R}^{d \times D}$ are linearly are linearly embedded to the image patch which is fed to the through the L Layers of the transformers to produce a set encoding vectors H_L.

2.3.2 Masked Image Modelling Head

The dataset of images $X = \{x_1, ..., x_N\}$ are also fed into image tokenizer[19] which converts them into $Z = \{z_1, ..., z_N\} \in V^{H \times W}$ tokens, where W denotes the width of the image, and H denotes the height, whereas V is the vocabulary which contains $V = \{v_1, ..., v_N\}$ discrete token indices.

2.3.3 Pretraining

We chose to utilize BeIT for domain adaptation because of its semi-supervised self-attention mechanism. The model consists of a 12-layer transformer with hidden size of 768 and 12 attention heads. The input image is resized into 224 x 224 resolution, anc converted into 14 x 14 image patches where each patch is of size 16 x 16. Simultaneously, the input image is also tokenized into 14 x 14 semi-tokens. 40% of the patches are randomly masked via blockwise masking algorithm and masked regions are attached with learnable embeddings. The patches are then fed into image transformer (encoder), where it produces the output, which is fed into masked image modelling head, which takes the input tokens and patches and tries to reconstruct the corrupted masked patch. The pre-training objective is to minimize the loss between original token and reconstructed token derived from the patched image. The loss function is be denoted by

$$\sum_{(x_i, \tilde{x_i} \in D)} (\mathbb{E}_{z_i \sim q_\phi(z|x_i)}[\log p_\psi(x_i|z_i)] + logp_\theta(\tilde{z_i}|\tilde{x_i})) \tag{1}$$

where $\mathbb{E}_{z_i \sim q_\phi(z|x_i)}[\log p_\psi(x_i|z_i)]$ denotes visual token reconstruction loss and $logp_\theta(\tilde{z_i}|\tilde{x_i})$ represents masked image modelling loss. Here x denotes the original image $\tilde{x_i}$ represents corrupted masked image and z denotes tokens obtained from the tokenizer; p is the pretraining objective which is maximize the log-likelihood of the corrupt visual tokens given a corrupted image and D represents the training dataset, $q_\phi(z|x_i)$ denotes the image tokenizer, $p_\psi(x_i|z_i)$ is the function that decodes original image from visual tokens and $p_\theta(\tilde{z_i}|\tilde{x_i})$ recover the visual image patches from corrupted patches.

2.3.4 Fine-tuning

After successful pretraining, a classification head (fully-connected network) is attached to the model, after which the model is fine-tuned on ImageNet-1k for another 500 epochs. For the first 400 epochs, the image size is the same as pre-training input (224 x 224), however, for the last 100 epochs, the images are reshaped and fed as having 384 x 384 resolution.

3 Experiment

3.1 Datasets

Due to the model size / training constraint in ViSDA 2021 Challenge [20], the training dataset provided was ImageNet-1k [7] which contains 1.4M images and 1000 classes. There were also 3 development datasets provided, as shown in Table 1. which were

1. ObjectNet [2] consisting of 50,000 images with 313 classes where only 113 classes are the same as the source dataset and the images are generally more difficult to classify due to the large differences in poses and backgrounds between each image of the same class.

2. ImageNet-R [21] which contains 30,000 images with 200 classes from the source dataset with varying visual styles and textures.

3. ImageNet-C [22] which is similar to ImageNet however the images are corrupted.

The development datasets weren't allowed to be used for training purposes. However other than that, they could be utilized for model development by tuning it's hyperparameters.

Dataset	Number of Images	Number of Classes	Note*
ImageNet(source)	1.4M	1000	
ObjectNet	50,000	313	Only 113 classes are the same as the source
ImageNet-R	30,000	200	Different texture/style
ImageNet-C	1.4M	1000	Corrupted

Table 1: Dataset Description

3.2 Training

For the experiment, only ImageNet-1k for training purposes. Even though ImageNet-C, ImageNet-R and ObjectNet were also allowed to be used for tuning hyperparameters from the pretrained source model, we in fact, didn't use it. Since our model was able to perform visual reconstruction on target visual token via self-supervised attention mechanism, we believed that it can also be able to learn various attributes from various images and that information could be cross-applied to different target datasets [18]. Moreover, we discovered that the model could also auto-seperate objects and classes without labelling. We

basically created a random and corrupted mesh of ImageNet-1k images with random noise and jittering to create a new class in existing dataset to tackle out-of-distribution classes in the target dataset. We do believe, however that pretraining self-supervised models on ObjectNet, ImageNet-C and ImageNet-R will increase the accuracy of the model.

3.3 Implementation Details

The model was pre-trained on ImageNet-1k. The augmentation used consisted of color-jittering, horizontal flipping and random resized cropping. We ran the pretraining for 500k steps with 1k batch size. The learning rate of 1.5e-3 and cosine learning weight decay of 0.05 is used. Adam optimizer with B1 = 0.9 and B2 = 0.999 is used. The training of 500k steps (1600 epochs) takes about 12 days using Nvidia Tesla V100 32GB GPU cards.

Methods	Parameters	ACC
ImageNet-1K(pretraining)	86M	82.4%
ImageNet-1K(pretraining + fine-tuning)	86M	83.0%

Table 2: Ablation Studies

The above table shows the top-1 accuracy trained on ImageNet-1k. The model achieves very high accuracy compared to various state of the art techniques [23, 8], which generally require larger datasets [24, 25].

3.4 Results: Performance on ViSDA 2021 Challenge

Our model ranked 1st place on the VisDA 2021 leaderboard,outperforming the second place by 15.04% on source-only accuracy and 7.73% on adapted model accuracy. To show that our model can outperform any existing domain adaptation techniques, we didn't use any development sets at all. Moreover, Table 3. shows that our adapted AUC is slightly lower than some of the other entries, that is truly understandable as the model was actually picking attributes from the source and applying them onto target sets rather than performing a max-mean discrepancies models which tries to reduce distance between source and target features. This clearly shows that our model is capable of picking attributes from the source and it can link those attributes from source-to-target in a completely self-supervised manner.

4 Future work

In this work, we've shown that pretraining transformers can successfully be applied for performing Domain Adaptation. In future, we would like to extend this technique to domain adaptation in multi-attribute object detection for

Methods	ACC(Adapted Model)	AUC(Adapted Model)	ACC(Source model)	AUROC(Source model)
babychick(ours)	56.29	69.79	56.29	69.79
liaohaojin	48.56	70.72	41.25	64.48
chamorajg	48.49	76.86	0.07	50.00
DXM-DI-AI-CV-TEAM	48.60	68.29	25.70	62.43
fomenxiaoseng	45.23	78.76	40.22	60.43

Table 3: Test Source results

successfully transferring the attributes from source to universal target dataset via textual embeddings

5 Conclusion

In this paper, we apply pre-train and finetune BeiT on ImageNet-1K and demonstrate that it is able to outperform current state-of-the-art domain adaptation techniques.

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: BERT pre-training of image transformers. *CoRR*, abs/2106.08254, 2021.

[2] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Joshua B. Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[6] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[8] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning, 2020.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] Geoffrey E Hinton, Zoubin Ghahramani, and Yee Whye Teh. Learning to parse images. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[11] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. *CoRR*, abs/1710.09829, 2017.

[12] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[13] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[14] Atif Belal, Madhu Kiran, Jose Dolz, Louis-Antoine Blais-Morin, Eric Granger, et al. Knowledge distillation methods for efficient unsupervised adaptation across multiple domains. *Image and Vision Computing*, 108:104096, 2021.

[15] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[16] Fabrizio J Piva and Gijs Dubbelman. Exploiting image translations via ensemble self-supervised learning for unsupervised domain adaptation. *arXiv preprint arXiv:2107.06235*, 2021.

[17] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8725–8735, 2020.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[19] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[20] Dina Bashkirova, Dan Hendrycks, Donghyun Kim, Samarth Mishra, Kate Saenko, Kuniaki Saito, Piotr Teterwak, and Ben Usman. Visda-2021 competition universal domain adaptation to improve performance on out-of-distribution data, 2021.

[21] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization, 2021.

[22] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[24] Christoph Schuhmann. 400-million open dataset, Oct 2021.

[25] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.