Investigating Self-Supervised Pre-training for Image Segmentation

Botond Branyicskai-Nagy

UCL, Department of Computer Science

1 Introduction

1.1 Background and Motivation

Image segmentation, the task of assigning a semantic label to each pixel in an image, is essential for autonomous driving, medical imaging, and remote sensing. Deep learning has fuelled remarkable progress in this field, alongside novel strategies for self-supervised (SSL) methods. The field transformed with the advent of FCNs in 2014 [13], followed by refinements with CRFs [2] to model spatial relationships between pixels. The next major breakthrough was U-Net [12]: skip connections combined high-resolution features increasing precision. DeepLab [3] achieved an increased receptive field, crucial for smaller objects. Recently, attention mechanisms have become central [8] with transformers' [15] capacity to learn long-range dependencies and relationships between features.

Early work focused on semi-supervised learning [14] and self-correction [5] demonstrating that segmentation models can benefit even from weaker forms of supervision. Recent work explores fully SSL methods for segmentation, using bootstrapping and novel modelling approaches [16]. The SSL approach is popular in many areas of machine learning, but especially so in computer vision tasks where we have access to vast amounts of unlabelled images, yet relatively small datasets with good quality labels for the specific sub-task at hand. Pre-training on a SSL task, such as Masked Image Modelling [17] – as inspired by successes in natural language processing – or Contrastive Learning [20] [18], allow us to leverage large datasets without targets for our downstream task, adding an intermediate goal that can be evaluated based on the inputs alone. With minimal architectural adjustments – simply swapping the prediction head – we can use most of the pre-trained model (its posterior weights) as initialisation for fine-tuning on our final task. As evaluated below, this can help with overall performance, especially if the fine-tuning set is small – particularly valuable in segmentation, where acquiring large labelled datasets is often costly.

Beyond these core areas, techniques such as classification as an intermediate fine-tuning step [1] and investigating how image resolution interacts with advanced pre-training techniques represent active topics in the field.

1.2 Aims

The experiments below aim to address three main topics, the first of which is the ability to overcome the limitations posed by a small labelled dataset for the seg-

Table 1: Encoder Configuration

Parameter	Image Size	Patch Size	Latent Dim.	Depth	Heads	MLP Dim.
Value	128	16	128	12	8	512

mentation task with self-supervision – specifically the SimMIM [17] framework. The degree to which pre-training aids the downstream segmentation task is investigated by comparing performance to a baseline model (with no pre-training). As others have shown, generic, global features are learned if the pre-train set is diverse and large enough, which can help bootstrap supervised training for a specific problem with similar inputs. To add to this, we also explore the effect of varying the unlabelled train set size, as this holds information about how reliant this procedure is on one of these desiderata – the large pre-train dataset.

As the second topic, the fine-tuning procedure is compared across a range of subset sizes to better understand where pre-training is most effective or helpful. The different pre-train sets are also used to help uncover some of the interaction between the two training stages in terms of their sizes.

Third, addressing the role of intermediate fine-tuning on a different task, specifically how using a dissimilar image dataset on a classification task affects the final segmentation performance. This topic ties in to the hierarchy of features extracted, as the new intermediate step steers the network to favour patterns at the level prominent in the dataset used. Therefore our contribution to previous work in this area is the quantified effect of an off-topical dataset from the main downstream goal: namely scenes and landscapes as opposed to pets. We consider this question in particular to be an interesting path for future work.

2 Methods

2.1 Self-Supervised Pre-Training: SimMIM

For the self-supervised pre-training step we use the recent, simple yet powerful framework for MIM [17]. At its heart, the process is built around a VisionTransformer [10], which requires splitting the image into patches (equal, mutually exclusive squares in our case) that are each passed through the encoder. As this method is adapted from language modelling, some of the same tricks are borrowed such as the cls token as an additional patch, as well as the added positional embedding providing information about the relative locations of the patches. The network is fed incomplete images where a large portion of the patches are randomly *masked*: replaced by a masking token, with the goal of reconstruction. This defines our self-supervised learning problem, as the target is part of the original image, against which the predictions are compared via the loss. Specific choices made within this framework for our experiments are below.

Our ViT encoder configuration in Table 1. totals 4M parameters, which can be considered very lightweight in comparison to even the standard vit_base_16 model of 86M. The main advantages of the 4M ViT are faster training times and

smaller saved model file sizes, traded off for reduced complexity and therefore limited richness of the features that can be learned from the images.

With simplicity as a guiding principle the decoder was chosen to be lightweight, a single linear layer mapping to pixel RGB values from the encoded patch tokens. This forces most of the complexity into the encoder, while achieving good performance. Following on from the findings of [17] a fairly large patch size (16×16) was chosen, as combined with a masking ratio of 50% (suggested range is 40-80%) this was shown to be an effective strategy. Optimisation using AdamW [7] proved to be the most reliable for our 100 epoch training loop minimising cross-entropy loss. A multi-step learning rate scheduler was employed with weight decay 0.05, a base learning rate of 10^{-4}, decreasing by a factor of 10 after epoch 50 and 85. The data was also augmented to minimise overfitting, with random rescaling, changing aspect ratio, horizontal flipping and colour jittering as suggested by the authors for the SimMIM procedure. The images are also normalised, resized and cropped to the input dimensions of the ViT used (128×128).

2.2 Fine-Tuning and Intermediate Classification

For segmentation, the architecture is in fact identical: we use the same encoder (Table 1), but the linear head is replaced as we are no longer mapping to full colours but rather to probabilities of the 3 classes for each pixel. Note that in this case the decoder has the same dimensions as the 3 values in the RGB encoding in masking aligns with the segmentation trimap. To investigate whether pre-trained ViT weights provide a better initialisation point for the segmentation model, we compare a baseline model to one initialised with the pre-trained weights. The baseline sees no pre-training at all, and is trained fully supervised on the segmentation task directly. Choosing cross-entropy loss with AdamW here again, we opted for a learning rate of $2 \cdot 10^{-3}$ and weight decay 10^{-4}, with a multi-step scheduler decreasing by 0.5 at steps 70, 90 and 95. We run for 100 epochs with dropout 0.1 in each experiment as this was found to be a good balancing point between performance before overfitting. We perform similar augmentations as before: normalisation, random horizontal flipping and colour jitter, along with the required resizing and cropping to the model's input dimensions.

For some experiments we introduce an additional intermediate fine-tuning step, where the linear head is replaced to output classification logits for 6 classes. This is then trained using AdamW with $8 \cdot 10^{-4}$ learning rate, decreasing with $\gamma = 0.1$ at epochs 180 and 190 out of 200, with 0.1 dropout. In addition to the augmentations in fine-tuning, we also employ MixUp [19] to boost performance.

3 Experiments

3.1 Data

As the standard choice for state-of-the-art computer vision pre-training [17][11] we used subsets of ImageNet-1K [4] for training with the SimMIM implementation, thus allowing for wider comparison. This is also motivated by ImageNet's

Fig. 1: SimMIM pre-training results: 4 examples of original, masked and reconstructed images from left to right.

free availability as a large collection of reasonable quality pictures encompassing a wide variety of object categories, encouraging diverse feature-learning that is relatively global and more likely to transfer well across different fine-tuning datasets. The range of randomly sampled subset sizes were 45k, 100k and 200k, as these provided well-differentiated performance while keeping to a reasonable training time of ∼4 hours on an NVIDIA RTX 4080 GPU. Seeing diminishing returns after the 200k datapoint we opted not to increase further, although we would expect some improvement in general, especially for much larger models.

The segmentation data of interest in the final fine-tuning stage is Oxford-IIIT Pet set [9] which includes 37 species of dogs and cats with ∼200 examples of each, totalling 7,349 images and their segmentation trimaps. For our purposes the pre-defined half train, half test division was ignored, and a smaller test set of 1k randomly selected images was chosen. This allowed for a more informative fine-tuning comparison to be carried out across a wider range of subset sizes.

For Intermediate fine-tuning, there is work already confirming the benefit of using ImageNet classification [17][1], however investigating how the choice of this intermediate dataset affects downstream segmentation performance remains – to the authors' knowledge – largely unexplored. To better understand the role of the contents and size of this dataset, we opted for a widely accessible, less similar dataset of intermediate size: Intel Image Classification [6]. This contains 14k training, 3k test images, 6 classes of scenes: buildings, forest, glacier, mountain, sea, and street – in stark contrast to the pet images in the final fine-tuning step.

3.2 Experimental Setup

We compare against the baseline model with the same architecture as that of fine-tuning, but all weights initialised randomly. This model only ever sees the pets dataset and its segmentation targets. The fine-tuned model however makes use of the encoder weights from the ImageNet pre-training and only has to finely adjust these, along with the new linear decoder head. Instances of both these models are then trained on the following subset sizes of the segmentation data: 250, 500, 1k, 2k, 4k and 6k. These portions were chosen to cover as wide a range as possible while retaining a reasonable hold-out test size.

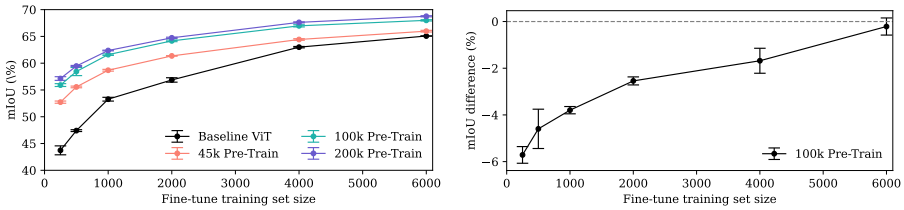

Fig. 2: (left) Mean IoU after fine-tuning across different dataset sizes and (right) the difference introduced by intermediate fine-tuning.

To explore the effect of different pre-train set sizes we repeated every fine-tuning experiment with the encoders pre-trained on 45k, 100k and 200k random images from ImageNet using the SimMIM procedure described above. We would expect more training data to provide an advantage, especially with less fine-tuning examples available, but this turns out not to be a trivial relationship.

The intermediate fine-tuning step was performed in a similar manner to the final segmentation training with the same subsets of the pets data, but with the added step in-between of the Intel Image Classification problem.

4 Results

Example results of SimMIM pre-training on ImageNet (200k) after 100 epochs are shown in Fig. 1. with predicted patches plotted in their true locations for clarity. Even at the 4M size the model captures the main edges and matches the overall colours of the masked patches remarkably. Some detail is lost, which we attribute to the constraints of this model size and compute limitations (100 epochs taking ∼4 hours). The default vit_base_16 configuration was also tested, but in the same time-frame the improvement in quality was deemed to be minor.

It is clear from the fine-tuning results in Fig. 2 that the number of images used – be that labelled or unlabelled – increases performance (in terms of the mean of the three classes' inverse over union metrics plotted, but also raw pixel accuracy). Both these behaviours see diminishing returns however, with the fine-tuning set hinting most prominently at a plateau after 5-6k images. The rate at which this occurs differs depending on pre-training size, with more generic information seen beforehand yielding convergence with less task-specific examples needed.

From no pre-training (baseline) to 200k, the improvement decreases quickly with each step up. The differences are most marked in the sparse fine-tuning data limit, where the algorithm is forced to rely mostly on pre-training, as it struggles to find features from the small pets subset alone. In this regime, previously learned representations such as generic shapes and edges are most helpful, becoming less and less relevant moving further towards a setting with ample information in the fine-tune set itself making pre-training less relevant because

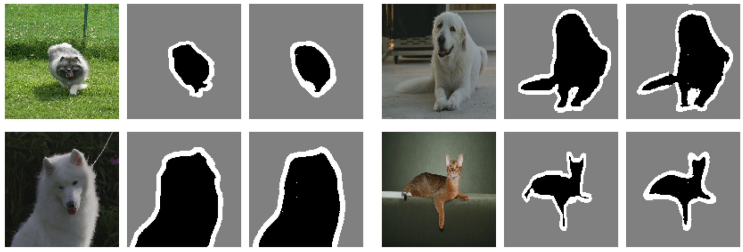

Fig. 3: Final fine-tuned model segmentation examples from Oxford-IIIT Pets.

the model has access to better, more specific data. For the top-performing segmentation model's results, we observe reasonable performance visually (see Fig. 3), the network capturing most features well, including contours.

With the introduction of intermediate fine-tuning, the mIoU is in fact reduced, as seen in the right hand side of Fig. 2. This goes against the findings of previous work where ImageNet was re-used in this step as well, which we postulate is due to the dissimilarity of useful patterns in the Intel scenes and pets. These images typically do not contain a single well-defined object and therefore may push optimisation in a sub-optimal direction for the downstream task. All of the tested pets subset sizes are in this negative-effect regime, however we see a clear trend in this gap decreasing, potentially beyond the zero line where it may become beneficial to perform this extra step. This could be explained by the added value in the additional intermediate data staying constant, while the optimiser's missteps are outweighed as the final dataset becomes overwhelmingly larger. The suboptimal convergence of the model on classification (85.50%, 87.16% and 87.77% with respective sizes) may also play a role – we attribute this to the limitations imposed by the small 4M model used.

5 Discussion

Limitations of SimMIM pre-training were identified for small encoders. Increasing pre-train size beyond 200k provided no improvements, however larger models may experience a different cut-off. As for intermediate classification, a comparison of different architectures and set of datasets with varying degrees of similarity would be required.

This project investigated several aspects of SSL applied to aid segmentation with limited examples. We demonstrate a significant dependence of its benefit on the amount and similarity of data used prior to fine-tuning. Considering the avenues of potential investigation mentioned, in summary future work in this area could help develop a more complete understanding of the exact behaviour of the many coupled variables involved.

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: BEiT: BERT Pre-Training of Image Transformers (Sep 2022), http://arxiv.org/abs/2106.08254, arXiv:2106.08254 [cs]
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs (Jun 2016), http://arxiv.org/abs/1412.7062, arXiv:1412.7062 [cs]
3. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (Apr 2018). https://doi.org/10.1109/TPAMI.2017.2699184, http://ieeexplore.ieee.org/document/7913730/
4. Deng, J., Dong, W., Socher, R., Li, L.J., Kai Li, Li Fei-Fei: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE, Miami, FL (Jun 2009). https://doi.org/10.1109/CVPR.2009.5206848, https://ieeexplore.ieee.org/document/5206848/
5. Ibrahim, M.S., Vahdat, A., Ranjbar, M., Macready, W.G.: Semi-Supervised Semantic Image Segmentation with Self-correcting Networks (Feb 2020), http://arxiv.org/abs/1811.07073, arXiv:1811.07073 [cs, stat]
6. Intel Corporation: Intel Image Classification Dataset, https://www.kaggle.com/datasets/puneet6060/intel-image-classification
7. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (Jan 2019), http://arxiv.org/abs/1711.05101, arXiv:1711.05101 [cs, math]
8. Mo, X., Chen, X.: Realtime Global Attention Network for Semantic Segmentation. IEEE Robotics and Automation Letters **7**(2), 1574–1580 (Apr 2022). https://doi.org/10.1109/LRA.2022.3140443, http://arxiv.org/abs/2112.12939, arXiv:2112.12939 [cs]
9. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
10. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image Transformer (Jun 2018), http://arxiv.org/abs/1802.05751, arXiv:1802.05751 [cs]
11. Peláez-Vegas, A., Mesejo, P., Luengo, J.: A Survey on Semi-Supervised Semantic Segmentation (Feb 2023), http://arxiv.org/abs/2302.09899, arXiv:2302.09899 [cs]
12. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (May 2015), http://arxiv.org/abs/1505.04597, arXiv:1505.04597 [cs]
13. Shelhamer, E., Long, J., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation (May 2016), http://arxiv.org/abs/1605.06211, arXiv:1605.06211 [cs]
14. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence (Nov 2020), http://arxiv.org/abs/2001.07685, arXiv:2001.07685 [cs, stat]
15. Tran, M., Vo, K., Yamazaki, K., Fernandes, A., Kidd, M., Le, N.: AISFormer: Amodal Instance Segmentation with Transformer (Mar 2024), http://arxiv.org/abs/2210.06323, arXiv:2210.06323 [cs]

16. Wang, Y., Zhuo, W., Li, Y., Wang, Z., Ju, Q., Zhu, W.: Fully Self-Supervised Learning for Semantic Segmentation (Feb 2022), http://arxiv.org/abs/2202.11981, arXiv:2202.11981 [cs]
17. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: SimMIM: A Simple Framework for Masked Image Modeling (Apr 2022), http://arxiv.org/abs/2111.09886, arXiv:2111.09886 [cs]
18. Zeng, S., Zhu, L., Zhang, X., Tian, Z., Chen, Q., Jin, L., Wang, J., Lu, Y.: Multi-level Asymmetric Contrastive Learning for Medical Image Segmentation Pre-training (Sep 2023), http://arxiv.org/abs/2309.11876, arXiv:2309.11876 [cs]
19. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond Empirical Risk Minimization (Apr 2018), http://arxiv.org/abs/1710.09412, arXiv:1710.09412 [cs, stat]
20. Zhao, X., Vemulapalli, R., Mansfield, P., Gong, B., Green, B., Shapira, L., Wu, Y.: Contrastive Learning for Label-Efficient Semantic Segmentation (Aug 2021), http://arxiv.org/abs/2012.06985, arXiv:2012.06985 [cs]