Improving Pixel-Wise Segmentation with Contrastive Pre-Training

Max Jappert

max.jappert.23@ucl.ac.uk

1 Introduction

Segmantic segmentation is a computer vision task that involves individually classifying the pixels of an image. Traditional approaches to solving this task include graph-based segmentation (Felzenszwalb and Huttenlocher, 2004) and utilising Markov random fields (Felzenszwalb and Huttenlocher, 2006). In recent years, the focus has shifted to using convolutional neural networks (CNNs), a supervised learning approach that tends to outperform more traditional models (Long, Shelhamer, and Darrell, 2015). It has been suggested that CNNs can be further enhanced by pre-training the model before supervised fine-tuning is performed, improving performance on various visual domains (Jiang et al., 2020) (Zoph et al., 2020). Pre-training using self-supervised contrastive learning in particular has shown promising results (He, X. Chen, et al., 2022) (Wang et al., 2021). Contrastive learning is a self-supervised learning algorithm which learns to distinguish between similar and dissimilar image pairs. By training on these relationships, the model learns the fundamental relationship structure of the data without relying on labels. It is thus capable of making use of an abundance of unlabeled data to learn rich representations which we can use to our advantage during a later supervised fine-tuning stage (Jaiswal et al., 2020).

Our minimum required project involves implementing and evaluating such a pre-trained deep learning model for semantic pixel-wise segmentation on the Oxford-IIIT Pet dataset (Parkhi et al., 2012). The dataset provides masks as labels, whereby each pixel is classified as either background, foreground or unknown. An example can be observed in Figure 4. We will compare the results of a model pre-trained on this dataset to a baseline model which was fine-tuned without pre-training in order to determine if pre-training improves the segmentation performance. Additionally, we will evaluate the difference between the test performance of the two models on different training dataset sizes.

For pre-training we use SimCLR, a simple framework for contrastive learning on a visual domain (T. Chen et al., 2020). Many contrastive learning frameworks have been proposed, such as MoCo (He, Fan, et al., 2020) and BYOL (Grill et al., 2020). We opted for SimCLR due to its relatively low computational requirements, whilst offering a good balance between simplicity and performance.

SimCLR works by utilising data augmentation and a contrastive loss function. The data augmentation is used to create multiple views of the same image, thereby constructing pairs of similar images. It does this by sampling two separate transform compositions $t \sim T$ and $t' \sim T$ from a distribution over a pool

of transforms T. This pool of transforms consists of cropping, flipping and color jitter. The pairs are used in conjunction with the contrastive loss function to learn similarity relationships.

For our open-ended question we propose extending SimCLR by adding three additional transforms to T. We evaluate how these changes influence the segmentation performance in order to gain a deeper understanding of the interaction between contrastive pre-training and segmentation.

2 Methods

SimCLR uses temperature-scaled cross entropy loss (NT-Xent) as the contrastive loss function. This loss function encourages the positive pairs (the augmented images) to be close in the learned latent space, while maximising the distance between negative (random, assumed to be dissimilar) pairs. A temperature scaling parameter τ determines the degree of separation NT-Xent encourages. For our experiments, we set $\tau = 0.5$.

We used 64×64 ImageNet data (Deng et al., 2009) for pre-training due to its availability and quality, thereby discarding the labels. We used the Oxford IIIT Pet Dataset (Parkhi et al., 2012) for fine-tuning, as assigned.

The model we use is based on a ResNet-34, a downgrade from the ResNet-50 proposed in the original SimCLR paper (T. Chen et al., 2020) due to limited computational resources. A ResNet is a deep CNN architecture incorporating residual blocks in order to avoid exploding or vanishing gradients, allowing for very deep and powerful networks (He, Zhang, et al., 2016). During pre-training, the ResNet's head is replaced by a pre-training head consisting of two fully connected layers with 512 neurons each and a ReLU activation function. After pre-training, we replace the pre-training head with the segmentation head and the loss function by a cross entropy loss. The segmentation head consists of five convolutional layers, with a bilinear upsampling layer between each pair. The segmentation head uses the latent space outputted by the head-less ResNet-34 and uses five alternating convolutional and upsampling layers to convert it to a segmentation mask.

We use the ADAM optimiser (Kingma and Ba, 2014) for both pre-training and fine-tuning, with initial learning rates $\eta_{PT} = 10^{-2}$ for the former and $\eta_{FT} = 10^{-3}$ for the latter. We use a step decay on the learning rate for pre-training with time-step $t = 10$ and decay rate $\gamma = 0.5$.

We use a 4:1 training/validation split for the ImageNet dataset. For the Oxford IIIT Pets dataset we use a 1:1 training+validation/test split as suggested by the authors and a 4:1 training/validation split on the former split. We use batch sizes of 2048 for pre-training and 128 for fine-tuning, the maximum our hardware allows for.

All computations were run on a computer with an RTX 4080 GPU with 16 GB VRAM, an Intel i7 13700K CPU, 32 GB RAM and a 2TB hard drive. The source code for this project can be found at `https://github.com/maxjappert/pet_segmentation`.

3 Experiments

For the minimum required project, we compare the fine-tuned model with a baseline model. The baseline model consists of the same architecture, differing only in that it has not been pre-trained, i.e., the fine-tuning starts from an uninformed state. We will consider the loss functions and the classification accuracy as metrics to evaluate model performance. Furthermore, we will consider Intersection over Union, which describes the ratio of the mask intersection and mask union between the predicted test masks and the ground truth.

We also evaluate how the performance of both models changes as the fine-tuning dataset size increases. This experiment stems from the intuitive notion that increasing the available data variety should increase the possibility of learning to correctly perform the task. We are interested in determining if the two models are differently influenced by this change.

For our open-ended question we will investigate how different combinations of new transformations added to T affect pre-training and the final model performance. The proposed transforms are elastic transformation, solarise and posterise. The visual effect of the proposed transforms can be observed in Figure 1. We argue that these transforms could increase the model performance as they fulfil the qualitative requirements by altering the image's appearance noticeably while retaining the semantic content. E.g., Figure 1 features four representations of what is clearly the same photo, while at the same time having obviously undergone four different transformations.

The second part of our open-ended question will involve using different dataset sizes for fine-tuning and evaluating

Fig. 1. Additional transformations proposed for the open-ended question. From left to right the figure shows the original, the elastic transformed, solarised and posterised images.

4 Results

The model achieves satisfactory pre-training performance, converging to a normalised pre-training loss of around 0.0003. This is an order of magnitude lower than the starting loss of 0.004. The training loss over the epochs is visualised in Figure 2.

Similarly, fine-tuning progresses as expected, whereby both the training and validation losses continuously decrease while both classification accuracies increase. This can be observed in Figure 3.

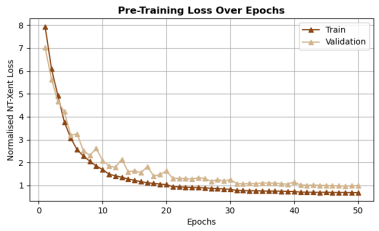

Fig. 2. Pre-training performance in terms of the normalized NT-Xent loss over the epochs.

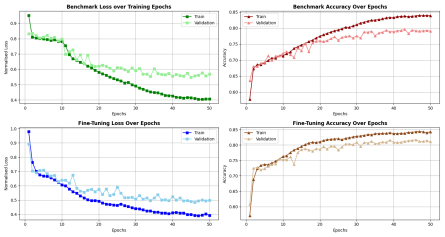

Fig. 3. Fine-tuning performance of the pre-trained and baseline models in terms of the normalized cross entropy loss and classification accuracy over epochs.

The training of the two models reveals that the pre-trained model provides better validation performance in terms of both the cross entropy loss and the segmentation accuracy when compared to the baseline model. This conclusion can be drawn from the plots in Figure 3, which provide a visualisation of the fine-tuning performance over epochs.

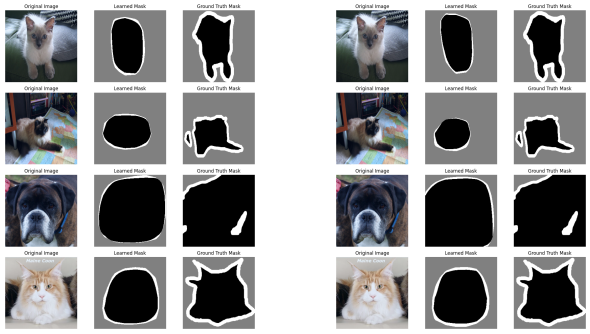

Fig. 4. Comparison of the learned masks on the test data using the baseline model for the left image and the pre-trained model for the right image. Each is visualised next to the corresponding ground truth mask and original images.

The pre-trained model achieves a test accuracy of 80.9%, compared to the 78.5% test accuracy of the baseline model. It achieved an IoU of 80.7% compared to an IoU of 80.8% achieved by the baseline model. The models learn masks which can be observed in Figure 4.

We can observe in Figure 4 that the model fundamentally does what it should. It recognises the area in the image which is in the foreground. Nonetheless, it does not do this in great detail, instead sketching a rough outline.

We observe that increasing the fine-tuning dataset size increases performance on the test set up to the 3669 images contained in the Oxford IIIT Pets training

dataset in terms of accuracy and IoU. Furthermore, we can observe a consistent difference between the baseline and the pre-trained models in terms of the performance metrics, whereby the latter consistently performs better. The results of this experiment are visualised in Figure 5.

The minimum required project results thus suggest that pre-training with SimCLR can at least marginally improve segmentation performance.

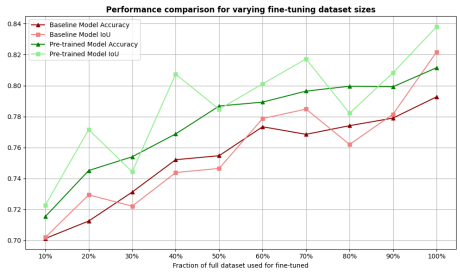

Fig. 5. Evaluated influence of the fine-tuning dataset size on segmentation test performance of both the pre-trained model.

Regarding the open-ended question, the results we obtain by introducing the new transforms suggest that their inclusion has no impact on the model performance in terms of none of the evaluated metrics. This can be observed in Table 1.

Table 1. *Oxford-IIT Pet* testset metrics for our fine-tuned models.

Model	**Accuracy**	**IoU**	**Dice Coefficient**	**Precision**	**Recall**
Baseline	0.785	0.807	0.813	0.879	0.939
SimCLR	0.809	0.808	0.812	0.930	0.927
ET+RP+RS	0.809	0.831	0.827	0.909	0.953
ET	0.812	0.844	0.812	0.923	0.961
ET+RP	0.812	0.823	0.821	0.924	0.940
ET+RS	0.809	0.824	0.822	0.915	0.946
RP	0.810	0.817	0.817	0.929	0.931
RP+RS	0.811	0.823	0.821	0.921	0.943
RS	0.812	0.829	0.825	0.917	0.948

5 Discussion

Regarding the open-ended question, our results suggests that as long as SimCLR performs transforms which render two different images while retaining their se-

mantic content, it will manage to learn equally useful representations. We thus argue that the exact choice of transforms is unimportant as long as the selection of transforms leads to useful results by noticeably altering the images while retaining their semantic content.

Regarding the second part of the open-ended question, while our results suggest that an increased training set size during pre-training leads to an increased segmentation performance, it is likely that this does not extend to arbitrarily large datasets. It would therefore be interesting to investigate the point at which this development stagnates or even diminishes due to overfitting.

It would be interesting to investigate how the downgrade from a ResNet-50 (in the original SimCLR paper (T. Chen et al., 2020)) to the ResNet-34 used in this project influences the pre-training and segmentation performances. This is particularly important as our model's performance is limited and there is a significant room for improvement. The current segmentation performance might be sufficient for some use-cases which do not require a detailed outline of the object in question, such as for identifying bounding boxes, yet it would not be sufficient, e.g., for automatically removing the background of an image.

Another approach to improving model performance could involve hyperparameter tuning, which could be done with a genetic algorithm and k-fold cross validation. The hyperparameters we could tune are τ, the batch sizes, the choice of optimiser and the learning rates η_{PT} and η_{FT}.

Finally, the numbers we present in our experiments are the result of a single execution. Our results would be more useful when aggregated over multiple executions and reported with a mean and standard deviation. We opted not to do this due to the computational cost of re-training multiple models.

6 Conclusion

The results of this project demonstrate that pre-training with contrastive learning can lead to an increased segmentation performance when compared to a baseline model which was fine-tuned without pre-training. The resulting pre-trained model performs well, but not exceptionally, with a pixel classification accuracy of around 81% on unseen data. This suffices for correctly detecting the rough outline of the animal in the foreground, but not for a detailed reconstruction. Furthermore, our results suggest that an increased dataset size during the contrastive learning phase leads to increased performance, up to a certain point. We also conclude that the inclusion of new, previously unexplored transformations into the SimCLR pre-training has no effect on segmentation performance. Overall, this project offers a solid and insightful foundation with plenty of room for improvement.

References

Chen, Ting et al. (2020). "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR, pp. 1597–1607.

Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Felzenszwalb, Pedro F and Daniel P Huttenlocher (2004). "Efficient graph-based image segmentation". In: *International journal of computer vision* 59, pp. 167–181.

— (2006). "Efficient belief propagation for early vision". In: *International journal of computer vision* 70, pp. 41–54.

Grill, Jean-Bastien et al. (2020). "Bootstrap your own latent-a new approach to self-supervised learning". In: *Advances in neural information processing systems* 33, pp. 21271–21284.

He, Kaiming, Xinlei Chen, et al. (2022). "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009.

He, Kaiming, Haoqi Fan, et al. (2020). "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738.

He, Kaiming, Xiangyu Zhang, et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Jaiswal, Ashish et al. (2020). "A survey on contrastive self-supervised learning". In: *Technologies* 9.1, p. 2.

Jiang, Ziyu et al. (2020). "Robust pre-training by adversarial contrastive learning". In: *Advances in neural information processing systems* 33, pp. 16199–16210.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2015). "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Parkhi, Omkar M et al. (2012). "Cats and dogs". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE, pp. 3498–3505.

Wang, Xinlong et al. (2021). "Dense contrastive learning for self-supervised visual pre-training". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3024–3033.

Zoph, Barret et al. (2020). "Rethinking pre-training and self-training". In: *Advances in neural information processing systems* 33, pp. 3833–3845.