Investigating U-Net Pretraining with SimCLR for Image Segmentation

Student Number: 18006555

Department of Computer Science, University College London,
66-72 Gower Street, London, WC1E 6EA, United Kingdom

1 Introduction

This paper explores the implementation of the U-Net architecture on self-supervised learning (SSL) for image segmentation on the Oxford-IIIT Pet dataset.

As of 2024, image segmentation is becoming increasingly popular in computer vision applications. The U-Net architecture has gained significant interest, due to its unique encoder-decoder structure, enabling it to effectively use both local and global information. Typically, supervised training of U-Net models requires large amount of labelled data, which can be expensive to obtain. We turn towards self-supervised learning as a promising alternative [3], which allows models to learn meaningful representations from unlabelled data via defined pretraining tasks.

In this paper, we focus on evaluating the Simplied Contrastive Framework for Contrasting Learning (SimCLR) [1] for pretraining U-Net architectures. SimCLR maximises agreement between differently augmented views of an image, offering robust visual representations. We chose SimCLR over other methods like BYOL or DINO due to its proven performance [1] and compatibility with U-Nets.

We compare SimCLR with Masked image Modeling (MIM) using Masked Autoencoders (MAEs). MIM predicts masked image regions to capture fine-grained contextual information. While SimMIM [2] applies MIM to vision transformers (ViTs), we extend this to U-Nets. We experiment with various masking techniques like random square patches and pixels.

We conclude with an open-ended investigation into architectural variations and training strategies of SimCLR, including skip connections, freezing encoder weights, and fine-tuning dataset size. Through these analyses, we aim to optimise SimCLR-based pretraining for U-Nets.

Our study aims to provide insights into SimCLR's effectiveness within U-Net architectures for image segmentation tasks, contributing to understanding self-supervised learning frameworks for such tasks.

2 Methods

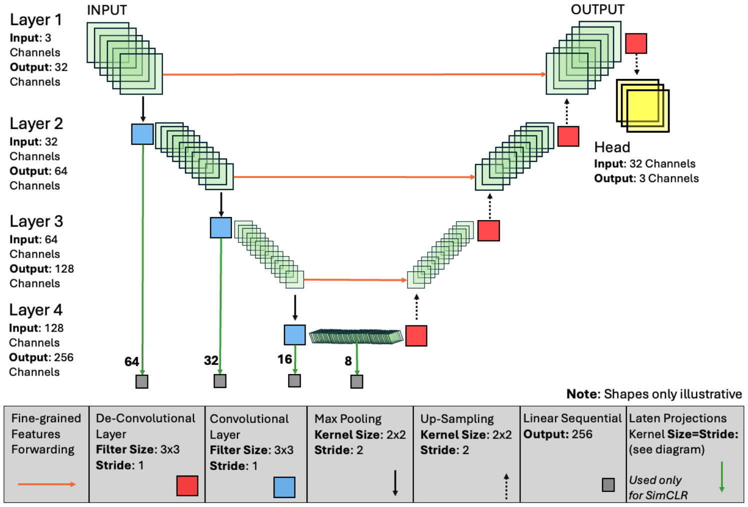
2.1 U-Net Hyperparameters

We used a U-Net CNN structure with an encoder-decoder architecture and skip connections. The encoder comprises four convolutional layers with 32, 64, 128, and 256 channels successively, mirrored in the decoder with transposed convolutional layers for upsampling. ReLU activation functions were applied after each layer for simplicity and training acceleration.

We initialised the learning rate at 0.001, adjusted dynamically with an Adam optimiser. Training spanned 20 epochs, for SimCLR and autoencoder pretraining, and Oxford-IIIT Pet dataset fine-tuning. Batch sizes were 384 for SimCLR pretraining and 256 for fine-tuning. Image sizes were 128×128 pixels for autoencoder and fine-tuning, and 64×64 pixels for SimCLR pretraining. We conducted experiments on PyTorch, utilising an NVIDIA GeForce RTX 3090 GPU.

During fine-tuning, skip connections were enabled to enhance gradient flow and segmentation performance. A temperature parameter of 0.2 was employed for the SimCLR contrastive loss function to scale similarity scores. We visualise this structure in Figure 1 below.

Fig. 1: U-Net Architecture Diagram


INPUT
OUTPUT
Layer 1
Input: 3 Channels
Output: 32 Channels
Layer 2
Input: 32 Channels
Output: 64 Channels
Layer 3
Input: 64 Channels
Output: 128 Channels
Layer 4
Input: 128 Channels
Output: 256 Channels
Head
Input: 32 Channels
Output: 3 Channels
64
32
16
8
Note: Shapes only illustrative
Fine-grained Features Forwarding
De-Convolutional Layer Filter Size: 3x3 Stride: 1
Convolutional Layer Filter Size: 3x3 Stride: 1
Max Pooling Kernel Size: 2x2 Stride: 2
Up-Sampling Kernel Size: 2x2 Stride: 2
Linear Sequential Output: 256
Laten Projections Kernel Size=Stride: (see diagram)
Used only for SimCLR

2.2 Datasets

For pretraining, we use the iNaturalist dataset [4], downsampled to 350,000 images of natural species (with plants and fungi removed). All images were downsized to a 128 × 128 pixel resolution, and further to 64 × 64 only for SimCLR pretraining, split into training and test subsets with a 80-20 ratio.

For fine-tuning, we use the Oxford-IIIT Pet dataset [5] containing 7,390 images of cats and dogs, alongside their corresponding tricolour segmentation maps. Additionally, all images were downsized to a pixel resolution of 128 × 128. Here we used a 90-10 split for training and test subsets.

2.3 SimCLR

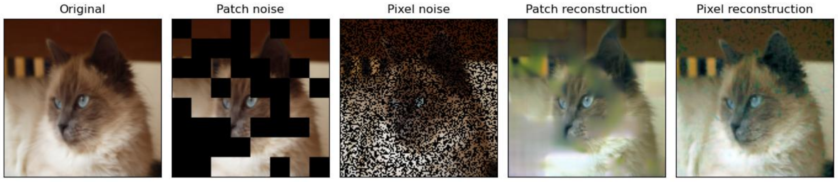
SimCLR is the contrastive learning strategy of choice to train the U-Net's encoder. The Normalised Temperature Cross-Entropy (NT-Xent) loss was used, which aims to maximise agreement between different augmentations of the same image (positive pairs), whilst penalising different images altogether (negative pairs). Here the low temperature parameter of 0.2 sharpens the distribution of similarity scores across the positive and negative pairs, emphasising larger differences between such pairs. The encourages the model to learn representations that are invariant to augmentations, as we idealise the model to learn the semantics of the image rather than focus on augmentations.

Regarding image augmentation, we opt for random resizes and crops, horizontal flips and colour jitter on the pretraining dataset, motivated by Chen at al. (2020)'s introduction of SimCLR [1]. Colour jitter was applied with parameters of 0.8 to Brightness, Contrast and Saturation, with a parameter of 0.2 for Hue [1]. Colour jitter is imperative to combat the model becoming heavily dependent on using specific patterns of pixel intensity values within images, hence reducing potential overfitting to such distributions.

2.4 Masked Autoencoder

We compared our contrastive learning technique of SimCLR with a Masked Autoencoder (MAE) where MIM is utilised by removing pixels from images and training the entire U-Net to reconstruct those images via inpainting. We evaluated varying masking techniques, including square patches 20 pixels wide and single pixel masks. In both instances, 75% of the image was removed and replaced with black pixels. We evaluate a third technique with no masking at all, and form a loss function computed by the mean squared error (MSE) between the original image and the image reconstruction for each of the three masking strategies. We observe an example reconstruction set in Figure 2 below.

Fig. 2: Image example of patch and pixel masking on an original image, alongside their respective reconstructions.



2.5 Supervised Fine-Tuning

For fine-tuning our SimCLR and MAE models for image segmentation, we turn to the Oxford-IIIT Pet dataset [4]. We retain the full dataset with no further image augmentations, and continue with 20 epochs and a batch size of 256 as noted in section 2.1.

We favoured the Dice loss function for fine-tuning, as demonstrated by Milletari et al. (2016) [6], which measures the similarity between predicted segmentation masks and ground truth masks by computing the overlap of their binary representations. Since segmentation tasks require precise delineation of objects, the Dice loss excels in capturing such fine-grained details and proves effective in this context.

2.6 Open-ended Investigation: Skip Connections, Frozen Encoders and Fine-tuning dataset size variations

In the U-Net architecture, skip connections are integral for maintaining spatial details during image segmentation tasks [7]. We compared two versions of U-Net: one with skip connections, adhering to the standard architecture where encoder features are merged with corresponding decoder features, and another without skip connections, thereby altering the architecture to solely rely on upsampled features from preceding decoder layers. This will allow us to evaluate the impact of skip connections in the autoencoder.

In U-Net, the choice of freezing encoder weights during fine-tuning is critical [8]. We compared freezing encoder weights versus allowing them to be updated during fine-tuning. When freezing encoder weights, we lock them during fine-tuning, preserving learned representations. We theorise this as beneficial since our fine-tuning dataset is relatively small and may result in overfitting and not allow our model to learn representations from scratch from the limited labelled data available. Exploring both frozen and unfrozen encoder weight configurations helps determine the optimal strategy for fine-tuning U-Net models for segmentation tasks.

We also evaluate the effect of varying the size of the fine-tuning dataset size, where we take subsets of the Oxford-IIIT Pet dataset [5] of 1%, 10% and our default of 90%. With smaller fine-tuning dataset sizes, the model's ability to learn from labeled examples is restricted. This can result in decreased performance, as the model may struggle to capture the variability and complexity of the task with limited training data. However, it also provides insights into the model's robustness and ability to generalise under data scarcity.

2.7 Evaluation Metrics

To assess segmentation performance, we considered the F1 score and the Jaccard index, analysing them per class, then averaging across classes, before finally averaging over batches. The Jaccard index [9], also knows as the Intersection over Union (IoU), measures the overlap between the predicted segmentation and the ground truth, indicating how well the model captures the spatial agreement between its predictions and the actual segmentation masks.

3 Experiments

With our methodologies outlined, we look towards formalising a series of experiments to investigate these approaches of self-supervised pretraining for image segmentation using U-Nets. We aim to answer the following topics:

3.1 SimCLR vs. Varying MAEs on segmentation performance

We described methods for two distinct techniques in this paper for pretraining a U-Net model: SimCLR and MAE. We look to pretrain 4 U-Net models for 20 epochs using the iNaturalist dataset [4], and then fine-tune each model for 20 epochs on 90% of the Oxford-IIIT Pet dataset [5]. The first U-Net uses SimCLR with skip connections and frozen encoder weights, whereas the remaining 3 U-Nets use a masked autoencoder approach with patch masking, pixel masking and no masking respectively, as described in section 2.4. We evaluate the segmentation performance via the F1 score and Jaccard index, as per section 2.7.

3.2 Open-ended Investigation: Skip Connections, Frozen Encoders and Fine-tuning dataset size variations in SimCLR U-Nets

Introduced in section 2.6, we run experiments to evaluate the impact of skip connections, freezing encoder weights and adjusting the fine-tuning dataset size within SimCLR U-Nets. We aim to cross-compare all three features and identify any dependencies. We pretrain and fine-tune 12 U-Net models for 20 epochs, split into free and frozen encoder weights, skip and no skip connection variants and fine-tuned on 1%, 10% and 90% of the total Oxford-IIIT Pet dataset [5]. We only consider the segmentation performances via the Jaccard index here, due to the number of models being compared and the sufficient correlation with the F1 score nonetheless.

4 Results

4.1 SimCLR vs Varying MAEs

Table 1: Pretraining: SimCLR vs Varying MAEs segmentation performance. Frozen encoder weights and 90% of fine-tuning dataset used.

Pretraining Method	**F1 Score**	**Jaccard Index**
SimCLR	0.775	0.650
MAE with Patch Masking	**0.779**	**0.656**
MAE with Pixel Masking	0.754	0.622
MAE with No Masking	0.748	0.614

The results of Experiment 3.1 indicate that a masked autoencoder applied with a patch masking technique provides the highest segmentation performance, via both F1 score and Jaccard index. However we note that this improvement is relatively minimal over SimCLR, with only a 0.516% and 0.923% difference in both metrics respectively, suggesting a largely insignificant difference between both approaches. Across solely MAEs, we identify patch masking to be the optimal technique above pixel masking, where the latter had a minimal increase over having no masking. We explain this since patch masking can capture larger contextual information within the masked regions, enabling more effective feature extraction and preserving spatial relationships [10].

4.2 Open-ended Investigation: Skip Connections, Frozen Encoders and Fine-tuning dataset size variations

Table 2: Cross-comparison of different SimCLR architectural adjustments: Frozen encoder weights, skip connections and fine-tuning subset size.

Encoder Weights	**Skip Connections**	**Fine-tuning Dataset Size**		
		90%	**10%**	**1%**
Unfrozen	No Skip	0.706	0.575	0.467
Unfrozen	Skip	0.716	0.602	0.448
Frozen	No Skip	0.469	0.650	0.550
Frozen	Skip	0.646	0.559	0.420

Table 2 demonstrates interesting results arising from a cross comparison of all 3 SimCLR architectural features. Zooming in on the choice of freezing encoder weights, with the default configuration of including skip connections on 90% of the fine-tuning dataset, we identify unfrozen encoder weights as the optimal approach with a 10.8% improvement. This effect is relatively consistent across even a reduced fine-tuning dataset size of 1% (6.67% improvement). We attribute this to the skip connections enabling the U-Net to leverage task-specific information from the decoder and incorporate it into the encoder during fine-tuning; frozen encoder weights would not use this task-specific feedback [7].

However when removing skip connections, we observe heightened differences with a 50.3% improvement from unfreezing encoder weights at 90% of the fine-tuning dataset (although we suspect this may be an anomalous result). Strikingly, we instead observe an improvement when freezing encoder weights at 10% and 1% of the fine-tuning dataset. Freezing the model weights prior to fine-tuning proves advantageous as we prevent the model from overfitting [7]. The lack of skip connections, coupled with frozen encoder weights, ensures that the model relies on pretrained representations for feature extraction.

Additionally, we observe a decrease in performance as the fine-tuning dataset size decreases, with a potential anomaly with a frozen encoder and no skip connections, where this trend does not follow the suggested pattern. This is largely expected as a smaller dataset will not cover and nor benefit from the full variability of the fine-tuning dataset.

5 Discussion

We observe interesting results from our experiments, most notably the intertwining of skip connections, frozen encoders and dataset size variations in our open-ended investigation. However, we note various limitations and future avenues of exploration. We were constrained to pretraining only on 64 × 64 resolution images, halved in density from the fine-tuning dataset resolution, due to computational resources. The potential anomaly of table 2 in section 4.2 would be further checked if time permitted.

Additionally, we did not include many further experiments conducted, for example varying the image resolution during inference to study the resolution-invariance of the U-Net model, comparing multi-level vs. bottleneck projection heads within the U-Net, and finally altering the choice of pretraining dataset from iNaturalist to other image sets (cats & dogs, inanimate objects, satellite imagery). These could have provided a more comprehensive analysis of SimCLR U-Nets for image segmentation, however were not discussed due to space constraints.

6 Conclusion

This paper explores self-supervised pretraining approaches for image segmentation, via SimCLR pretraining on a U-Net architecture. We conclude that SimCLR with skip connections and a frozen encoder performs similarly to a masked autoencoder, which has its own optimal setup using patch masking. Strong performance is observed by SimCLR applied to a U-Net CNN, evaluated by F1 score (0.775) and Jaccard index (0.650), indicating its suitability and novelty via its fully convolutional nature.

Our open-ended investigation highlights the benefits of freezing encoder weights of the U-Net when the fine-tuning dataset size is insufficient, risking overfitting and an inability to learn from diverse features. Including skip connections (assuming a sufficiently large fine-tuning dataset) is best complemented with unfrozen encoder weights, allowing the U-Net to use task-specific information from the decoder. Without skip connections, we turn towards frozen encoder weights when our fine-tuning dataset is small.

This paper sets the foundations for analysing SimCLR pretraining on U-Net CNNs for image segmentation tasks. We set a future outlook of extending this exploration via the aforementioned resolution-invariance, projection head variants and the semantic proximity of pretraining and fine-tuning datasets.

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A Simple Framework for Contrastive Learning of Visual Representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020)
2. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: SimMIM: A Simple Framework for Masked Image Modeling. arXiv preprint arXiv:2109.03238 (2021)
3. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data. In: Proceedings of the 24th International Conference on Machine Learning, pp. 759-766 (2007).
4. inaturalist21. (2021). iNaturalist 2021 competition dataset. Retrieved from `https://github.com/visipedia/inat_comp/tree/master/2021`
5. Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). Cats and Dogs. Oxford: Visual Geometry Group, University of Oxford. Retrieved from `https://www.robots.ox.ac.uk/~vgg/data/pets/`
6. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 565-571 (2016). doi:10.1109/3DV.2016.79
7. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, pp. 234-241. Springer, Cham (2015).
8. Tran, T.Q., et al.: Fine-tuning Convolutional Neural Networks for Biomedical Image Analysis: Actively Frozen or Passively Thawed? In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 528-536. Springer, Cham (2019).
9. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440 (2015).
10. Litjens, G., et al.: Patch-Based Image Analysis: A Review and Comparison of Methods. IEEE Reviews in Biomedical Engineering **11**, 187-200 (2018).