# Self-Supervised Pre-Training with Masked Auto-Encoder For Blood Image Semantic Segmentation

Ming Hill[1]

[1]Faculty of Computer Science, Dalhousie University

September 16, 2024

**Abstract**

# 1 Introduction

# 2 Background

# 3 Related Work

# 4 Methods

This section outlines the approach used for data processing, model configuration, pre-training, and fine-tuning. We used a Vision Transformer Masked-AutoEncoder (ViTMAE) pre-trained on unlabeled blood images which was used on downstream the task for semantic segmentation with a UNETR architecture.

## 4.1 Data Processing

**ViTMAE** The dataset used to pre-train our VITMAE consisted of 23,040 16 x 256 x 256 (channel, height, width) images. We split the data randomly into train (80%), validation (10%) and test (10%) images. Within the dataloader, we split the 16 x 256 x 256 images into 16 x 64 x 64 sub-images, creating 16 new images per image, resulting in 294,912 train images and 36,864 validation/test images. Data augmentation done in pre-training consisted of random vertical flips, random horizontal flips, and we normalized our data. We used a batch size of 16, and 32 which when split resulted in batch sizes of 256 and 512 respectively.

**UNETR** Our dataset for semantic segmentation consisted of 2,382 (16 x 64 x 64) unlabeled blood sample images along with 2,382 (64 x 64) identical images that were sparsely labeled. Only 224,711 pixels were labeled out of the total 9,756,672 total number of pixels. The labels consisted of 10 total classes:

| **Category** | **Number of Pixels** |
|---|---:|
| Background | 80166 |
| White Blood Cell | 7442 |
| Platelet | 1472 |
| Outer Red Blood Cell | 14506 |
| Inner Red Blood Cell | 2422 |
| Beads | 749 |
| Monster Beads | 737 |
| Sensor Scratch | 4021 |
| Chamber Top Scratch | 1069 |
| Debris | 62161 |
| Bubble | 49966 |

Table 1: Pixel count for various categories

The dataset was split into train (80%), validation (10%) and test (10%) sets with the same random state for both labeled and unlabeled images so that the splits were the same. The six total datasets were zipped with their corresponding train/valid/test images and saved as an npz file. We created a custom UNETR dataset to store labeled and unlabeled images and passed them into data loader simultaneously. For our train data-loader, we included random horizontal and vertical flips, using random seeds to ensure both images were augmented the same. We also normalized the unlabeled data images. No augemenstion was applied to validation and test data loaders but normalization was done on unlabeled images. Batch sizes were set to 32 for both training and validation sets.

## 4.2 Model Configuration

**VITMAE** We took inspiration from "Masked Autoencoders Are Scalable Vision Learners." The model we used was one of the Hugging Face transformer models.

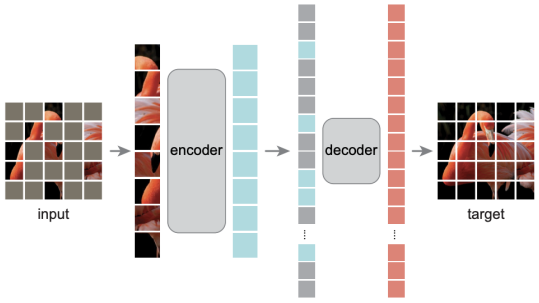


Figure 1: Diagram showing the encoder-decoder process

We mainly used 2 model configurations excluding patch size and mask ratio. Model 1 was a symmetric model with the following configurations:

| **Parameter** | **Value** |
|---|---|
| NORM_PIX_LOSS | 1 |
| LAYER_NORM_EPS | 1e-6 |
| HIDDEN_SIZE | 192 |
| INTERMEDIATE_SIZE | 768 |
| NUM_HIDDEN_LAYERS | 6 |
| NUM_ATTENTION_HEADS | 6 |
| HIDDEN_DROPOUT_PROB | 0.0 |
| ATTENTION_PROBS_DROPOUT_PROB | 0.0 |
| DECODER_HIDDEN_SIZE | 192 |
| DECODER_INTERMEDIATE_SIZE | 768 |
| DECODER_NUM_HIDDEN_LAYERS | 6 |
| DECODER_NUM_ATTENTION_HEADS | 6 |

Table 2: Configuration Parameters for Model 1

The second model was a smaller asymmetric model.

| **Parameter** | **Value** |
|---|---|
| NORM_PIX_LOSS | 1 |
| LAYER_NORM_EPS | 1e-6 |
| HIDDEN_SIZE | 192 |
| INTERMEDIATE_SIZE | 512 |
| NUM_HIDDEN_LAYERS | 4 |
| NUM_ATTENTION_HEADS | 4 |
| HIDDEN_DROPOUT_PROB | 0.0 |
| ATTENTION_PROBS_DROPOUT_PROB | 0.0 |
| DECODER_HIDDEN_SIZE | 128 |
| DECODER_INTERMEDIATE_SIZE | 256 |
| DECODER_NUM_HIDDEN_LAYERS | 2 |
| DECODER_NUM_ATTENTION_HEADS | 2 |

Table 3: Configuration Parameters for Model 2

These models were run with a (2 x 2), (4 x 4) and (8 x 8) mask patch size. Mask patch size represents how many pixels are masked out in each patch. We also had a mask coverage ratio of 50%, 75%, and 90%. These ratios represent the percent of which the image is completely masked.

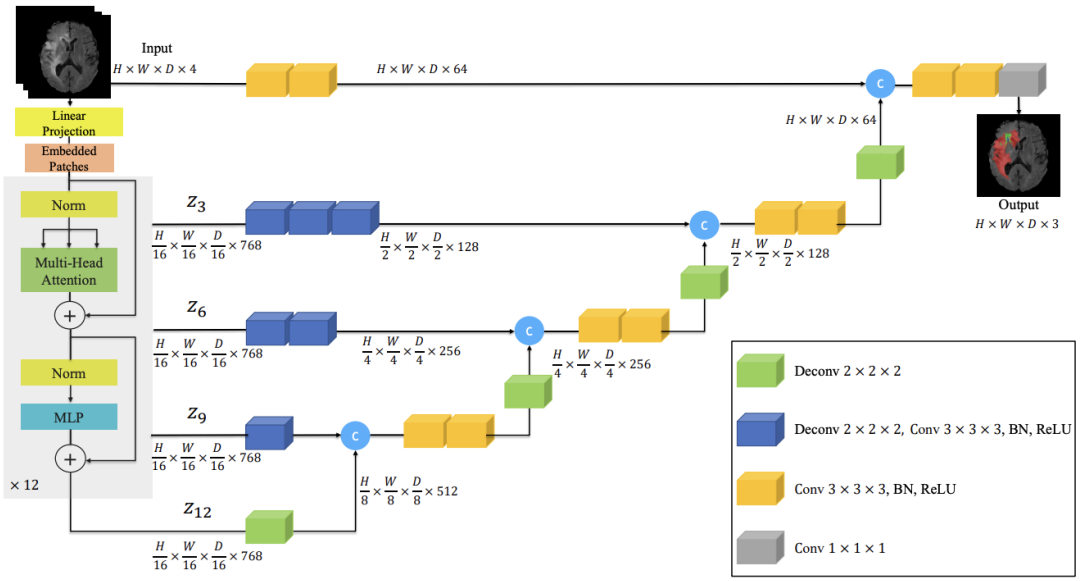


Figure 2: Overview of UNETR architecture

**UNETR** Our UNETR model is based off the paper "UNETR: Transformers for 3D Medical Image Segmentation." The UNETR (Unet Transformer) is a U-shape architecture similar to a UNET. The encoder side of the model is a transformer while convolutions are used on the decoder side. The architecture implements skip connections from layers of the encoder to layers of the decoder.

The encoder is directly taken from our pre-trained ViTMAE model. First, we initialized a ViTMAE configuration with the correct parameters for the specific model. We also initialize a ViTModel configuration with the same parameters. We then create a new ViTMAE Model with the given configuration and load the saved state dict into the new model. We then extract the ViT encoder from the ViTMAE, create a new ViT Model with the correct parameters, and load the state dict from the extracted encoder.

We used the MONAI framework, a free open-source library designed to help build the decoder side of the UNETR. We built a total of 3 different architectures. A UNETR for our (2 x 2), (4 x 4), and (8 x 8) mask patching. Depending on the size of the mask size, the UNETR has more or less skip connections. For 8 x 8, we had skip connections on layers 3 and 5 as well as connections from input and output. For 4 x 4, our skip connection was on layer 4 and the input and output. For 2 x 2, we only had a skip connection from the input and output layer. When transferring outputs from the vit encoder to the skip connections, we removed a single pixel from the hidden states.

For the UNETR, we had a feature size of 32, 64, 128, and 256, with spatial dimensions of 2 and 16 in-channels.

## 4.3 Evaluation Metrics

**ViTMAE** For our ViTMAE, we used MSE (mean squared error) between the reconstructed image and the original image. The loss function was supplied through HuggingFace within their ViTMAE class.

**UNETR** For semantic segmentation, we want to identify the loss between multi-class labeling, therefore we used a Cross Entropy Loss. We ignored index 255 which were the unlabeled pixels in the labeled dataset. We also used metrics precision, recall, accuracy and F1 score for individual classes and a multi-class confusion matrix.

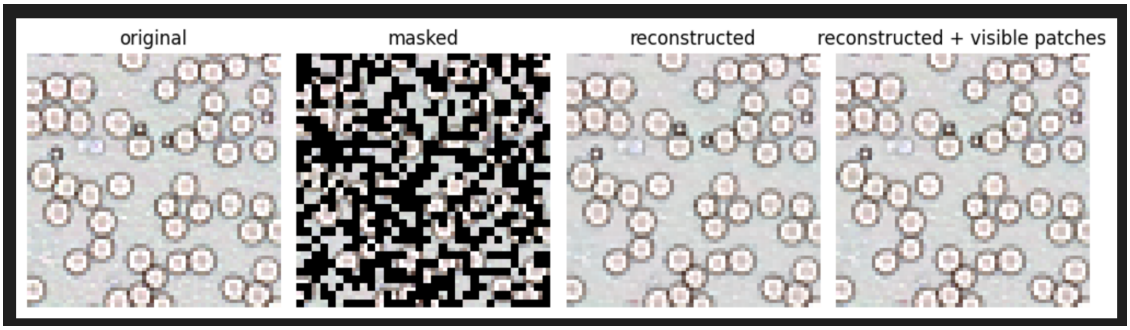


Figure 3: ViTMAE Reconstruction with Model 1, (2 x 2) mask patch, 50% mask ratio

### 4.4 Optimizers

**VITMAE** For the VITMAE, we used a Adam optimizer with BETA values of (0.90, 0.95), and 0.0 weight decay and 1.0e-5 learning rate.

**UNETR** For the UNETR architecture, we used a Adam optimizer with BETA values of (0.90, 0.95), and 0.0 weight decay and 1.0e-3 learning rate.

## 5 Results

### 5.1 VITMAE

**Model 1 vs Model 2** Comparing Model 1 with Model 2, model 1 had significantly better performance around the board. When working with a patch size of 4 and masking ratio of 50%, running on 100 epochs for Model 1 and 50 epochs for Model 2, model 1 resulted with a loss of 0.375 and Model 2 resulted in a loss of 0.577. The reconstructions for both were subpar, however, Model 2's reconstruction had unusual color values of green and yellow.

**Patch Sizes** During this project, we mainly ran experiments involving patch sizes of (2 x 2), (4 x 4) and (8 x8). Due to the nature of the data we were using, it was evident that using a (16 x 16) patch size, as mentioned in the ViTMAE paper, would be too large and cover entire objects. With a patch size of (2 x 2), red blood cells were masked halfway. Running a (2 x 2) on Model 1, with 50 epochs and masking ratio of 50% took 1160 minutes. The results of a (2 x 2) for loss were 0.34 with a near-perfect reconstruction. With Model 2, the loss was 0.37 after 50 epochs and a masking ratio of 50%. The reconstruction was not as clear as Model 1, however, it did have the correct color scheme as the original input photo. (8 x 8) patch size were used in with Model 1 with 100 epochs and a masking ratio of 50%. The loss came out to 0.47 however the reconstructions were not consistent and generally performed poorly when compared to (4 x 4) and (2 x 2) patches.

**Masking Ratio** In our project, we tried 3 separate masking ratios on our best-performing pre-trained models. This includes a 50%, 75%, and 90% mask ratio. All experiments were performed with a 2-patch size on Model 1 and 50 epochs. A 50% mask ratio produced a loss of 0.34, 75% ratio produced a loss of 0.40 and a 90% had a 0.52 loss. These experiments were run to confirm our choice of using a 50% mask ratio.

### 5.2 UNETR

**UNETR (4 x 4)** The pre-trained model for our (4 x 4) UNETR model had the following configurations. It was Model 1 with a (4 x 4) mask patch size, 50% mask ratio, 0.001 learning rate run for 100 epochs. We tried 4 different feature sizes which included 32, 64, 128, and 256 run on either 300 or 500 epochs. We ran 500 on earlier experiments and noticed loss would level out at around the 200-250 epoch range, which is why

we lowered the total epochs to 300. When looking at metrics for semantic segmentation, we focused on the F1 score of plates and loss.

| Feature Size | Loss | Platelet F1 |
|---|---|---|
| 32 | 0.107 | 0.89 |
| 64 | 0.085 | 0.90 |
| 128 | 0.093 | 0.90 |
| 256 | 0.103 | 0.93 |

Table 4: Metrics for Different Feature Sizes with 4 x 4 patch size

**UNETR (2 x 2)** The pre-trained model used for a (2 x 2) model had the following configuration trained on Model 1 with a (2 x 2) mask patch size, 50% and 75% mask ratio, learning rate of 0.001 run for 50 epochs. We ran 3 different feature sizes, each with a different number of epochs.

| Feature Size | Epochs | Loss | Platelet F1 |
|---|---|---|---|
| 64 | 300 | 0.105 | 0.84 |
| **128** | **150** | **0.063** | **0.93** |
| 256 | 200 | 0.223 | 0.92 |

Table 5: Performance Metrics for Different Feature Sizes and Epochs

**UNETR (8 x 8)** The pre-trained model for (8 x 8) UNETR had the configurations with model 1 consisting of (8 x 8) patch size, 50% mask ratio, a learning rate of 0.001 and run for 100 epochs. In our experiments, we ran with feature sizes 32, 64, 128 and 256. All experiments were run for 500 epochs.

| Feature Size | Loss | Platelet F1 |
|---|---|---|
| 32 | 0.184 | 0.63 |
| 64 | 0.21 | 0.56 |
| 128 | 0.116 | 0.82 |
| 256 | 0.11 | 0.89 |

Table 6: Performance Metrics for Different Feature Sizes

# 6 Discussion

## 6.1 Self-Supervised vs Supervised

We compared our model with a UNET model completely trained on supervised data. The UNET was trained on the same dataset that we fine-tuned our UNETR on and tested on the same dataset.

From the experiments, the best model was a UNETR (2 x 2) with a feature size of 128. The full results are below compared with a supervised trained UNET. The UNET model had a training loss of 0.08, however the platelet's F1 scores were significantly worse compared to the pre-trained model. However, other than the platelets, the other objects seem to have similar F1 scores.

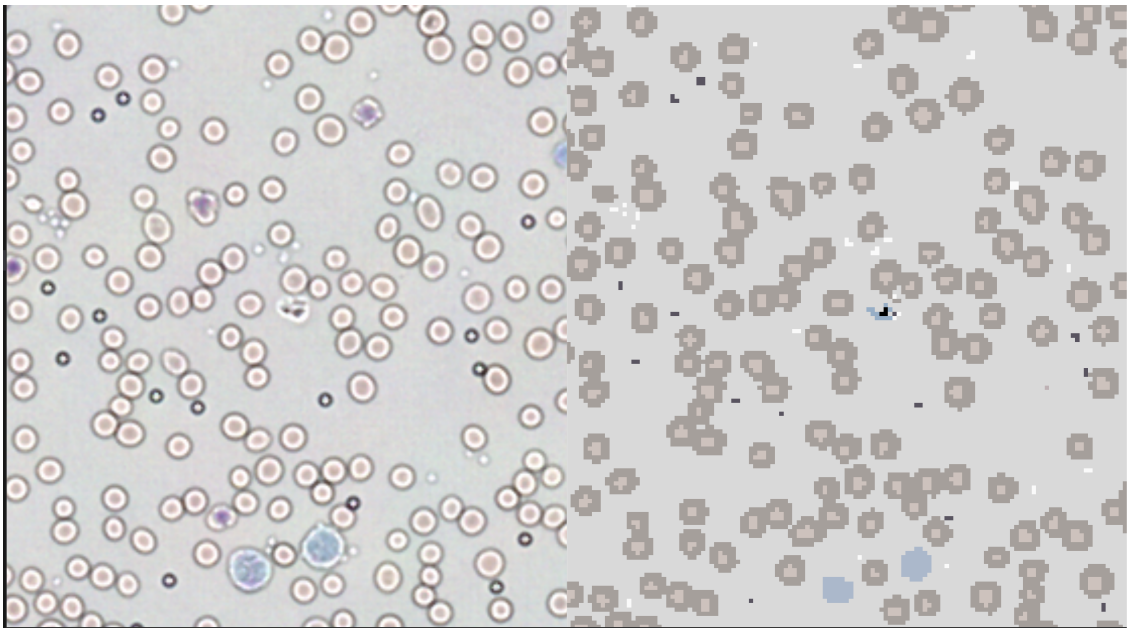

Figure 4: Segmentation Mask with best UNETR model

# 7 Conclusion

# References


1. Hatamizadeh, A., Yang, D., Roth, H., et al. "UNETR: Transformers for 3D medical image segmentation". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 574–584.
2. He, K., Chen, X., Xie, S., et al. "Masked autoencoders are scalable vision learners". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16000–16009.
3. Zhou, Z., Sodha, V., Pang, J., et al. "Self pre-training with masked autoencoders for medical image classification and segmentation". *arXiv preprint arXiv:2111.09807*, 2021.
4. Consortium, M. *MONAI: Medical Open Network for AI*. https://monai.io. Accessed: 2024-07-26. 2020.

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Background | 0.99 | 0.99 | 0.99 | 6960 |
| White Blood Cell | 0.97 | 0.99 | 0.98 | 940 |
| Platelet | 0.93 | 0.92 | 0.93 | 135 |
| Outer RBC | 0.97 | 0.96 | 0.97 | 737 |
| Inner RBC | 0.99 | 0.81 | 0.89 | 155 |
| Beads | 1.00 | 0.94 | 0.97 | 62 |
| Monster Bead | 0.00 | 0.00 | 0.00 | 0 |
| Sensor Scratch | 0.96 | 0.96 | 0.96 | 131 |
| Chambertop Scratch | 0.98 | 0.89 | 0.94 | 132 |
| Debris | 1.00 | 0.98 | 0.99 | 6360 |
| Bubble | 0.96 | 1.00 | 0.98 | 3211 |
| **Accuracy** | 0.98 | | | |
| **Macro Avg** | 0.89 | 0.86 | 0.87 | 18823 |
| **Weighted Avg** | 0.98 | 0.98 | 0.98 | 18823 |

Table 7: Class Report of UNETR with self-supervised pre-training and supervised fine-tuning

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Background | 0.99 | 0.98 | 0.98 | 6960 |
| White Blood Cell | 0.96 | 0.97 | 0.97 | 940 |
| Platelet | 0.78 | 0.80 | 0.79 | 135 |
| Outer RBC | 0.94 | 0.97 | 0.96 | 737 |
| Reticulocyte (Missing) | 0.00 | 0.00 | 0.00 | 0 |
| Inner RBC | 0.92 | 0.79 | 0.85 | 155 |
| Beads | 0.95 | 0.84 | 0.89 | 62 |
| Monster Bead | 0.00 | 0.00 | 0.00 | 0 |
| Sensor Scratch | 0.88 | 0.90 | 0.89 | 131 |
| Chambertop Scratch | 0.91 | 0.97 | 0.94 | 132 |
| Debris | 1.00 | 0.99 | 0.99 | 6360 |
| Bubble | 0.99 | 1.00 | 0.99 | 3211 |
| **Micro Avg** | 0.98 | 0.98 | 0.98 | 18823 |
| **Macro Avg** | 0.78 | 0.77 | 0.77 | 18823 |
| **Weighted Avg** | 0.98 | 0.98 | 0.98 | 18823 |

Table 8: Class Report of UNET with supervised training

# Supplemental Material