Segmentation in Art Images

Priyanshu Arora p1arora@ucsd.edu
Morgan Lafferty mlaffert@ucsd.edu
Katya Katsy kkatsy@ucsd.edu
Rafferty Chen rac003@ucsd.edu

1. Problem Statement

The cross depiction problem describes the difficulty neural nets have in correctly identifying objects when depicted in different styles. This problem often arises when applying such networks to solve computer vision tasks in artwork as the major vision datasets are comprised primarily of realistic photographs.

In the paper "Improving Object Detection in Art Images Using Only Style Transfer" [3] the authors attempt to address this issue by taking advantage of stylization techniques. They demonstrate that style transfer is an effective method to improving upon state of the art object detection in artwork by fine-tuning a Faster R-CNN on a dataset created by AdaIn style transfer and highlighting its strong performance in detecting people.

For our project, we will be extending upon their efforts by exploring whether style transfer is effective for semantic segmentation of people as well. By doing so, we hope to expand the applications of their work and provide further confidence that style transfer on existing datasets is an effective alternative to laboriously annotating mass amounts of paintings for the same task.

2. Method

2.1. Datasets

For training and validation we use the styleCOCO dataset constructed in [3]. This is a subset of the original COCO dataset containing over 60,000 images that depict people and has been stylized to look like artwork via AdaIn style transfer.

Our test set contains 200 internet-sourced pieces of artwork which we annotated using Roboflow, a publicly available platform for developing computer vision applications.

2.2. Stylization

AdaIn style transfer [2] is a neural network that takes two input images and applies the style of one to the content of the other. The styleCOCO dataset uses the Painter by Numbers dataset from Kaggle as the source of stylized images.

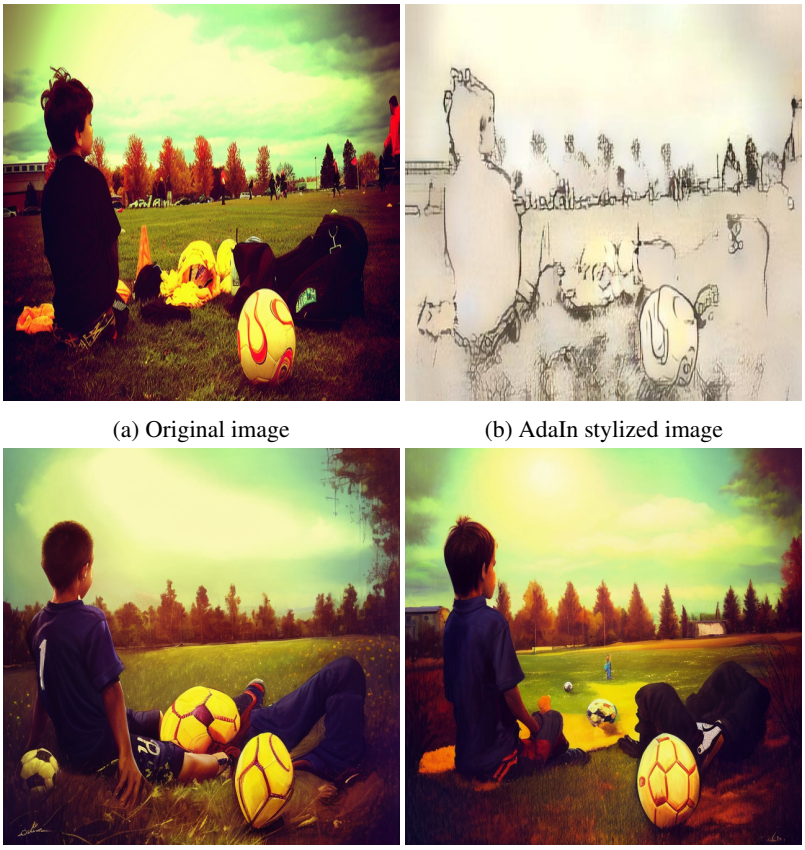
Figure 1: Examples of AdaIn style transfer

In addition to the AdaIn style transfer method, we experimented with Stable Diffusion [4] as a potential alternative. One advantage that this would have over the AdaIn style transfer method would be that Stable Diffusion models were trained on many works of art and would be better at creating an image that looked like a painting, with recognizable backgrounds and human figures. In Figure 2, we can see how Stable Diffusion creates images that are much more recognizable as paintings than the AdaIn stylized image. However, it also modifies the images in such a way that the original masks cannot be used. In Figure 2c, the boy now has his arms visible and his legs out in front of him and the people in the background were removed. In Figure 2d, the boy's arms are in a different position, the people in the background were removed, and there is a new person wearing a light blue jersey in the background. Because of these changes made to the original images, we decided not to pursue Stable Diffusion as our method of stylization.

2.3. Models

2.3.1 UNet

U-Net is a fully convolutional neural network originally developed for biomedical image segmentation. The U-Net architecture is characterized by a U-shaped design, hence its name; It consists of an encoder path and a decoder path.



(a) Original image (b) AdaIn stylized image

(c) Stable Diffusion stylized image (d) Stable Diffusion stylized image

Figure 2: Examples of outputs from Stable Diffusion

In the encoder path, the input image is progressively downsampled through convolutional and pooling layers, which capture high-level features and spatial information. This helps extract abstract representations of the image. The decoder path then performs upsampling and convolutional operations to gradually reconstruct the original image resolution. Skip connections are also introduced, which allow the decoder to combine high-level features from the encoder path with low-level features to better localize and refine the segmentation boundaries.

Because we are only segmenting people, there is only 2 classes (person and background), and therefore we use Binary Cross Entropy Loss. We also add dice score in order to prevent the model from predicting the background class only.

We train the U-Net model on the training set of 60,000 images and 3,000 images for validation set.

2.3.2 DeepLabv3+

DeepLabv3+ [1] is a Deep CNN (DCNN) model created by Google in 2018, which was constructed specifically to address two of the common pitfalls encountered in DCNN semantic segmentation: reduced feature resolution caused by pooling and scale invariance. DeepLabV3+ employs spatial pyramid pooling within an encoder-decoder architecture, which allows the network to learn multi-scale features in the encoder component and recovers spatial information in the decoder component.

An image is fed into a pre-trained Deep CNN; in our implementation, we used ResNet50. The output of a higher level layer of the pre-trained CNN goes through atrous spatial pyramid pooling: the module performs convolution at different dilation rates, the output of the spatial pyramid and pooling are combined, convoluted, and passed along to the decoder. The decoder takes in the output of a lower level layer of the pre-trained CNN, convolutes it, then combines it with the upsampled decoder output. The result of the concatenation is then convoluted and upsampled, producing a segmentation mask.

Atrous convolution makes it possible to use a pre-trained network to extract denser feature maps by removing downsampling operations from the last layers and upsampling the corresponding filter kernels. Thus, one can control the res-

olution at which features are computed within the DCNN component without having to learn extra parameters.

DeepLabv3+ uses two methods to add scale invariance: an image pyramid and spatial pyramid pooling. An image pyramid extracts features from multiple scales of input, allowing objects at different scales to be captured by different feature maps. Spatial pyramid pooling examines feature maps with filters and pooling operations at multiple rates and multiple effective field-of-views, allowing the model to capture objects at multiple scales.

Like UNet, we train DeepLabv3+ on the cocoStyle training set of 60,000 images and validation set of 3,000 images.

2.3.3 SegFormer

SegFormer [5] is a Transformer-based architecture recently developed by NVIDIA for semantic segmentation tasks that introduces novel redesigns in both the encoder and decoder components. It uses a positional-encoding-free and hierarchical Transformer encoder pre-trained on ImageNet-1k that is able to generate both fine and course features from the input images. In addition, it employs a lightweight All-MLP decoder design that effectively aggregates information from the different encoder layers, combining both local and global attention to produce high-quality segmentation masks. Training is performed using cross-entropy loss. When evaluated on three well-known semantic segmentation datasets (ADE20K, Cityscapes, and COCO-Stuff), SegFormer set a new state-of-the-art in terms of efficiency, accuracy, and robustness.

Due to computational resource limitations, for our project we perform fine-tuning of the SegFormer model on a random subset of 10,000 styleCOCO images with an 80-20 training-validation split. We then evaluate model performance on our full custom art dataset with respect to mIOU as done in the original SegFormer paper [5].

3. Results

3.1. UNet and DeepLabv3+

After several attempts at hyperparameter turning, both the U-Net and DeepLabv3+ architectures were experimentally unsuccessful at segmentation. Both models consistently converged to predicting all pixels as the background class, which was quantitatively equivalent to at most a 0.5 dice coefficient.

For U-Net, one possible reason for this is that the model may be sensitive to initialization of model parameters, as the skip connection can amplify any errors in the initial weights. This can make it more difficult to train U-Net compared to other architectures. Another main reason is that U-Net was trained from scratch, not pretrained on any dataset such as ImageNet. Pre-training on large datasets such as ImageNet may enable our models to understand abstract features that are useful for our task as well.

styleCOCO	
# Features	Test mIOU
128	0.72
256	0.79
300	0.77
400	0.79

Figure 3: Results for SegFormer model trained on styleCOCO. "Num Features" is the feature embedding size. "Test mIOU" is the mean Intersection Over Union on the test set.

COCO	
# Features	Test mIOU
128	0.66
256	0.71
300	0.71
400	0.70

Figure 4: Results for SegFormer model trained on normal COCO. "Num Features" is the feature embedding size. "Test mIOU" is the mean Intersection Over Union on the test set.

With respect to DeepLabv3+, while the ResNet model within in the encoder was pretrained on ImageNet, it seems that the architecture was still not sophisticated enough to learn the complex distribution of human outlines during the process of training. Training for a larger amount of epochs did not seem to provide significant improvement in performance. Either the pre-trained component of DeepLabv3+ was not able to generalize to our target of stylized human outlines, or our task simply requires a better, transformer-based architecture to be performed well.

3.2. SegFormer

The SegFormer model outperformed the other two models on a smaller subset of the training dataset. Figure 3 shows the mean Intersection Over Union for the test dataset, considering the feature embedding size. Empirically, the model exhibits improved performance with higher encoder feature counts, enabling better capture of crucial high-level features for accurate predictions by the MLP. Future research can focus on optimizing the number of features required, as performance tends to decline beyond a certain threshold.

Figure 4 shows the mIOU for the SegFormer model trained on normal COCO images. As expected, the performance of the SegFromer model trained on normal COCO images is far poorer than SegFormer trained on styleCOCO.

Additionally, Figures 5a and 5f demonstrate that the Seg-

Former model tends to perform well when the people are depicted as the main focus of the art piece and is able to segment the human figures with strong precision. The model also appears to be able to locate many instances of people within the same piece of work as seen by 5a in which it locates the singular woman to 5c where it captures all nine people. On the hand, the model tends to struggle when the artwork includes objects obstructing parts of the human form as seen in 5b where a man is holding a guitar in front of his body as well as when there is an abstract representation of a human face or body as in 5e. This is likely because style transfer is unable to mimic the more abstract art styles.

Overall, the SegFormer models shows very promising results for the task of human segmentation in artwork, especially since it was trained on a relatively small (10,000) number of images, and provides confidence that style transfer techniques may be an effective strategy for addressing the cross depiction problem. We hypothesize that performance would only improve if trained on the full styleCOCO dataset of 60,000+ images and leave such experimentation to future work.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. 2

[2] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017. 1

[3] David Kadish, Sebastian Risi, and Anders Sundnes Løvlie. Improving object detection in art images using only style transfer. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 1

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 1

[5] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021. 3

(a) (b) (c)

(d) (e) (f)

Figure 5: SegFormer results on test images