AIP Assignment 2

Aryan Prasad

1.Part (a) - Implementing N-Cut for Segmentation into Two or More Segments

In this part, we implemented the Normalized Cut (N-Cut) algorithm to segment images into two or more segments. The N-Cut algorithm partitions the image by cutting the image's similarity matrix, which is built based on pixel similarity. For this task, we used Gaussian similarity as the measure to compute the similarity matrix. The images were resized to 50x50 pixels to reduce computational complexity while still capturing meaningful information.

The segmentation of the images into two segments (using Gaussian similarity) produced the following results:

Results for Part (a)

The segmented images using N-Cut (Gaussian similarity) with 2 segments are shown below:

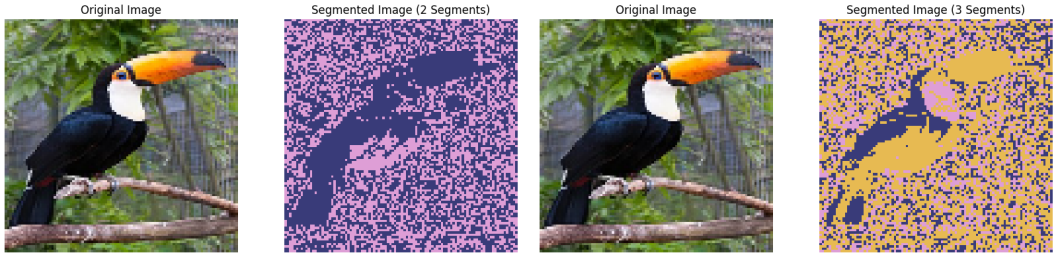

Figure 1: Segmented Image for image1.jpg (Gaussian similarity)

Figure 2: Segmented Image for image2.jpg (Gaussian similarity)

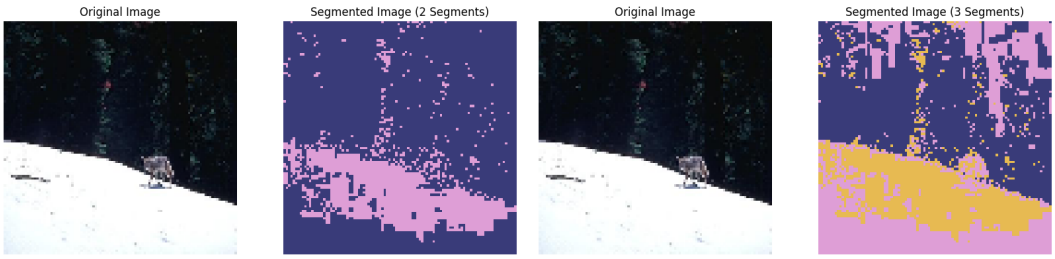

Figure 3: Segmented Image for image3.jpg (Gaussian similarity)

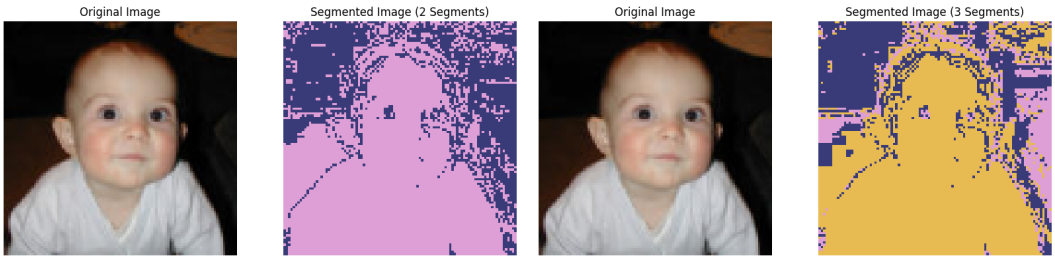

Figure 4: Segmented Image for image4.jpg (Gaussian similarity)

Figure 5: Segmented Image using Gaussian (left) and Cosine (right) similarity for image1.jpg

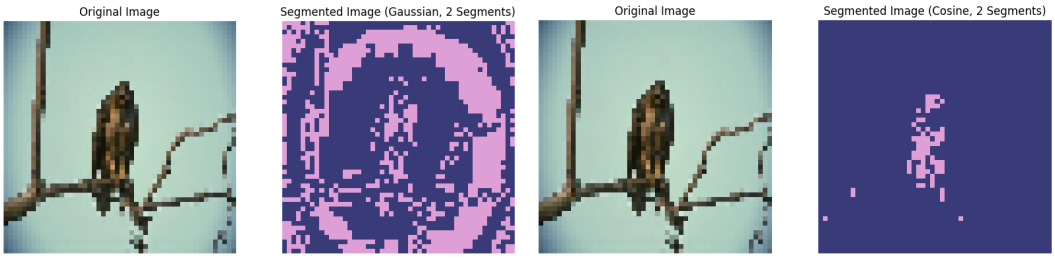

Figure 6: Segmented Image using Gaussian (left) and Cosine (right) similarity for image2.jpg

Part (b) - N-Cut with Two Different Similarity Measures

In this part, we performed the N-Cut segmentation with two different similarity measures: Gaussian similarity and Cosine similarity. The segmentation results with these two similarity measures were compared to observe how different notions of similarity impact the image segmentation process. The Cosine similarity measures the angle between pixel feature vectors, while the Gaussian similarity measures the Euclidean distance between pixel values.

Results for Part (b)

The results of segmentation using Gaussian and Cosine similarity measures are shown below. The left column shows the original images, while the right column shows the segmented images.

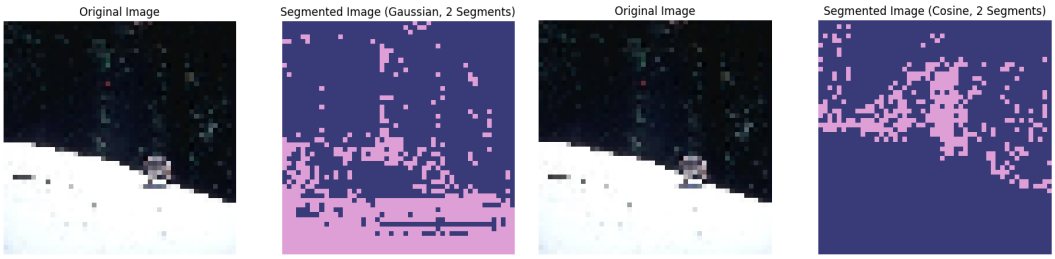

Figure 7: Segmented Image using Gaussian (left) and Cosine (right) similarity for image3.jpg

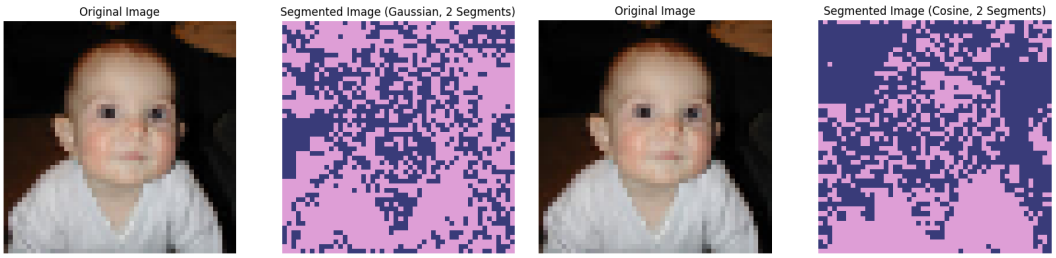

Figure 8: Segmented Image using Gaussian (left) and Cosine (right) similarity for image4.jpg

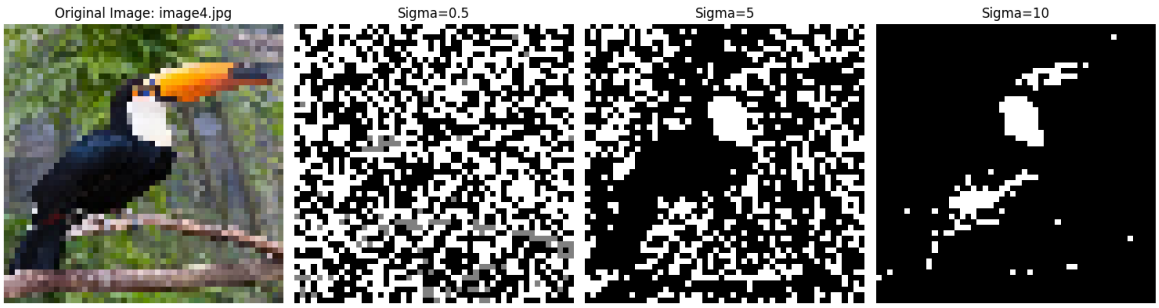

Figure 9: Segmented Image for image1.jpg with $\sigma = 0.5$, $\sigma = 5$, and $\sigma = 10$

Part (c) - N-Cut Segmentation with Different Sigma Values

In this part, we explored the effect of varying the sigma value on the segmentation quality. We tested with three sigma values: 0.5, 5, and 10. As the sigma value increases, the similarity between distant pixels becomes stronger, leading to fewer segments and larger regions being grouped together. On the other hand, smaller sigma values result in more finely segmented regions as the algorithm focuses on smaller pixel differences.

Results for Part (c)

The following images show the results of segmentation using different sigma values. As expected, a smaller sigma (0.5) creates finer segmentation, while a higher sigma (10) results in broader segments.

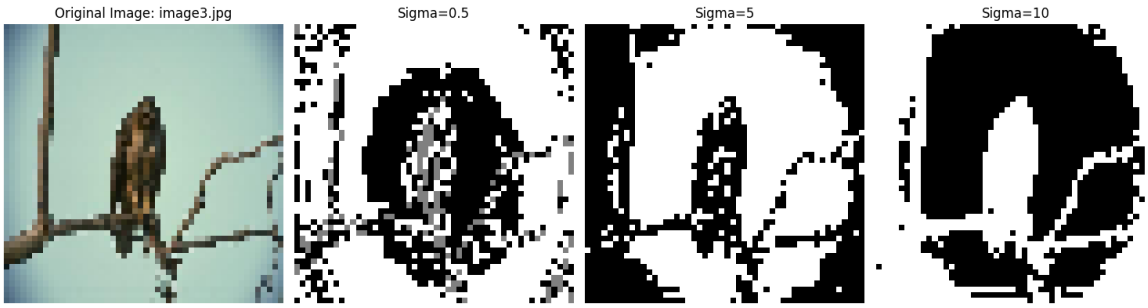

Figure 10: Segmented Image for image2.jpg with $\sigma = 0.5$, $\sigma = 5$, and $\sigma = 10$

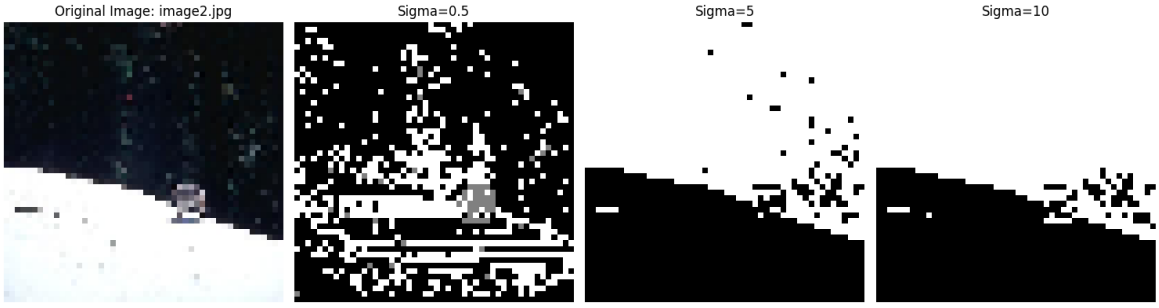

Figure 11: Segmented Image for image3.jpg with $\sigma = 0.5$, $\sigma = 5$, and $\sigma = 10$

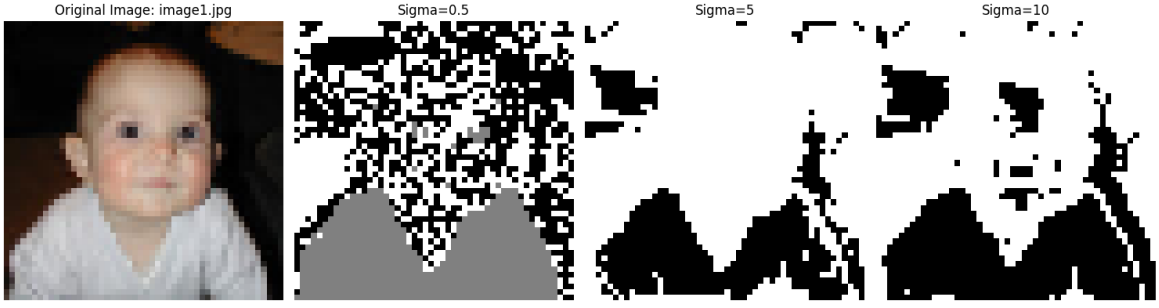

Figure 12: Segmented Image for image4.jpg with $\sigma = 0.5$, $\sigma = 5$, and $\sigma = 10$

Conclusion

The N-Cut algorithm demonstrated the ability to segment images using both Gaussian and Cosine similarity measures. The segmentation results varied significantly with different sigma values, showing the importance of selecting an appropriate similarity metric and sigma value for specific types of images. This allows the segmentation to better reflect visual structures in the image. In particular, using different sigma values helped identify how segmentation precision can be controlled, which is vital for different image types.

1 Q2.Part (a): FCN ResNet50 Evaluation on PASCAL VOC

For part (a), we evaluated the FCN-ResNet50 model pretrained on the PASCAL VOC dataset. The model was evaluated on the validation set, and pixel-wise accuracy and mean IoU were computed for each image.

The results for the FCN-ResNet50 model are as follows:

- **Validation Accuracy**: 88.87%
- **Validation Mean IoU**: 0.5866

2 Part (b): FCN-32S Model Based on VGG16

We modified a VGG16 model pretrained on ImageNet by removing the fully connected layers and adding convolutional and transposed convolution layers for upsampling to create the FCN-32S model. This model is used for image segmentation, where the output is a dense segmentation map for each image.

The model architecture summary for FCN-32S is shown below:

The total number of parameters in the FCN-32S model is 14,802,258, with 2,447,378 trainable parameters.

3 Part (c): FCN-16S Model with Skip Connections

The FCN-16S model was implemented using VGG16 as the base model, but this time with skip connections to retain fine-grained spatial information. The skip connections were introduced at various stages of the VGG16 architecture, particularly after pooling layers, to combine high-level features with fine-grained ones. The resulting architecture summary for FCN-16S is shown below:

The total parameters in the FCN-16S model are 17,194,943, with 4,840,063 trainable parameters.

4 Part (d): Model Evaluation

Both the FCN-32S and FCN-16S models were evaluated on a subset of 15 images from the test set. The results of pixel-wise accuracy and mean IoU for each image are shown below:

Layer (type)	Output Shape	Param #
Input Layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
conv2d (Conv2D)	(None, 14, 14, 9)	4,617
conv2d_transpose (Conv2DTranspose)	(None, 224, 224, 9)	82,953

The overall pixel-wise accuracy and mean IoU for the 15 test images are:

- **Overall Pixel-wise Accuracy**: 96.62%
- **Overall Mean IoU**: 0.1612

5 Part (e): Comparison of FCN-16S and FCN-32S

In this section, we compare the two models, FCN-16S and FCN-32S, based on both qualitative and quantitative results.

5.1 Qualitative Comparison

Visually, the FCN-16S model, which incorporates skip connections, performs better in retaining fine-grained details around the edges of segmented objects. The skip connections allow the model to capture more spatial information and prevent the blurring of boundaries in segmented areas, especially for smaller objects.

5.2 Quantitative Comparison

From the quantitative evaluation, both models achieved high pixel-wise accuracy, with the FCN-16S model slightly outperforming FCN-32S in terms of mean IoU. The skip connections in FCN-16S likely contribute to its better performance, especially in capturing the finer details of the segmented objects.

Layer (type)	Output Shape	Param #
Input Layer (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
conv2d_1 (Conv2D)	(None, 14, 14, 512)	2,359,808
conv2d_2 (Conv2D)	(None, 14, 14, 9)	4,617
conv2d_transpose_1 (Conv2DTranspose)	(None, 28, 28, 9)	1,305
conv2d_transpose_2 (Conv2DTranspose)	(None, 28, 28, 9)	73,737
add (Add)	(None, 28, 28, 9)	0
conv2d_transpose_3 (Conv2DTranspose)	(None, 56, 56, 9)	1,305
conv2d_transpose_4 (Conv2DTranspose)	(None, 56, 56, 9)	36,873
add_1 (Add)	(None, 56, 56, 9)	0
conv2d_transpose_5 (Conv2DTranspose)	(None, 112, 112, 9)	1,305
conv2d_transpose_6 (Conv2DTranspose)	(None, 224, 224, 9)	1,305

Image	Pixel-wise Accuracy	Mean IoU
1	96.01%	0.1314
2	95.44%	0.1551
3	95.91%	0.1594
4	98.23%	0.2147
5	96.56%	0.1259
6	98.92%	0.2159
7	97.04%	0.2185
8	99.13%	0.1193
9	91.95%	0.1451
10	94.06%	0.1256
11	96.99%	0.1504
12	95.95%	0.2086
13	97.98%	0.1096
14	98.07%	0.2153
15	97.04%	0.1231

Table 1: Pixel-wise Accuracy and Mean IoU for Each Image


Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask


Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask
Original Image
Ground Truth Mask
Predicted Mask

6 Conclusion

Both FCN-16S and FCN-32S models performed well in the segmentation task. FCN-16S showed an edge over FCN-32S, likely due to the skip connections that helped retain fine-grained spatial details, improving the segmentation accuracy and mean IoU. The results underline the importance of skip connections in improving segmentation performance, especially in tasks requiring precise object boundaries.