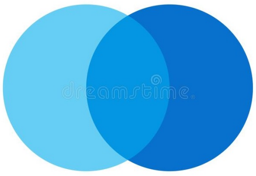
E9 246 — Advance Image Processing

Name: Alokendu Mazumder **Department and Course:** PhD (EE)
Student Number: 20134 **Link for Code(s):** Click here

Problem 1 **Implement nCut algorithm (2 segments). Analyze for 2 different similarity measures.**



(a) A simple figure

(b) Train 1 from PASCAL

(c) Train 2 from PASCAL

(d) Bird from PASCAL

Figure 1: Images used to test NCut for 2 segments

Implementation details: All images are taken from PASACL dataset except that circle one. I have downsampled each image to (50,50) dimension as it takes alot of time to compute similarity matrix of large dimensional images. Even if images of dimension above (150,150) are taken, computation of similarity matrix take huge amount of time. Two similarity measures are used, the first one is:

$$w_{ij} = \begin{cases} e^{\left(\frac{-\|F(i)-F(j)\|^2}{\sigma_I^2}\right)} e^{\left(\frac{-\|X(i)-X(j)\|^2}{\sigma_X^2}\right)} & \text{if } \|X(i)-X(j)\|^2 < r \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

here, $X(i)$ and $X(j)$ are position of i_{th} and j_{th} node corresponding to pixel location in a flatten vector. $F(i)$ and $F(j)$ are pixel brightness value at i^{th} and j^{th} node. I have fixed the threshold value to 50 units and tested different images for two different distance measures. I borrowed the idea of second measure from the family of heavy tailed distribution, **Weibull distribution's PDF** is used as the second measure with parameters ($\lambda = 1$ and $k = 0.5$).

$$w_{ij} = \begin{cases} \frac{1}{2\|X(i)-X(j)\|+\epsilon} e^{-\|X(i)-X(j)\|} * e^{\|F(i)-F(j)\|^2} & \text{if } \|X(i)-X(j)\|^2 < r \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

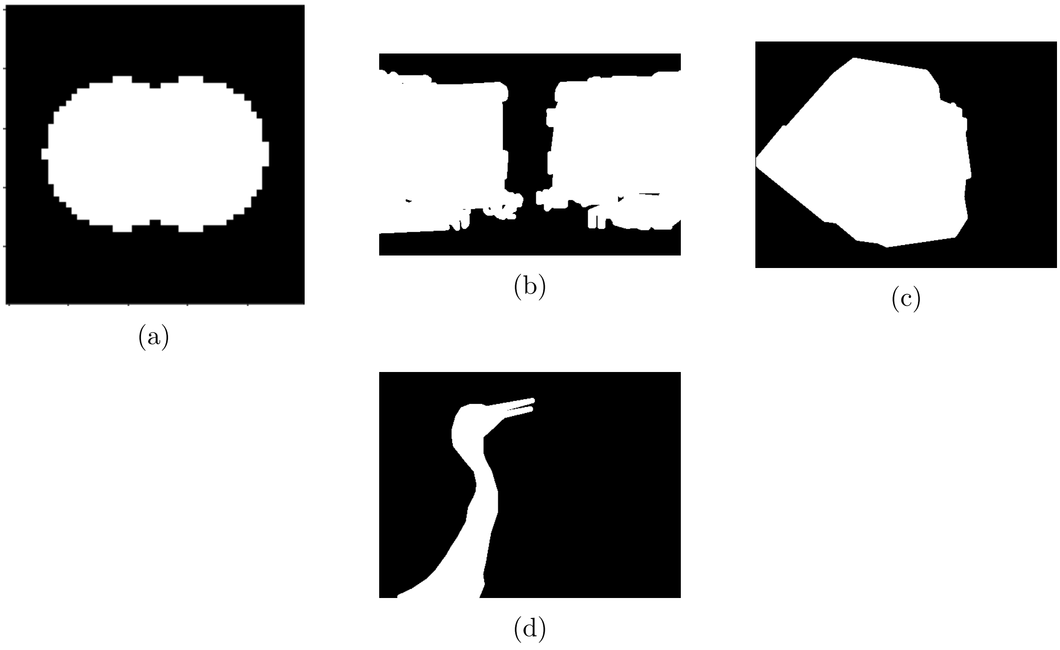


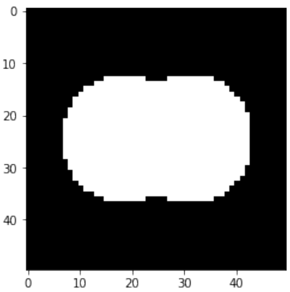
Figure 2: Ideal mask of the above images

A small corrector term ϵ is introduced into denominator to avoid blowing up when it's zero. Value of ϵ is taken to be 0.1.

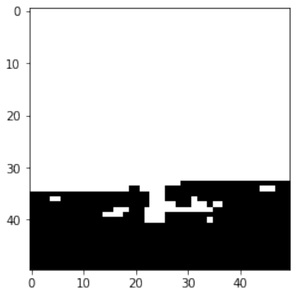
Results and Observations: It's observed that similarity measure (2) gives better or similar results as compared to measure (1). Quantitative results over the two similarity measures are being computed in terms of Mean intersection over union (MIOU) score. Reason for slightly better performance of similarity measure (2) can be its **heavy tail** nature, i.e the function decays a little slow as compared to Gaussian. If there a little increase in distance the similarity doesn't falls off rapidly as compared to Gaussian kernel. There may be other reason also, but I have written what I can infer from my knowledge about functions and graphs. I have used one synthetic image, the circle one, both measures are performing well. But in complex images, they don't perform upto the mark, however, if you can see segmentation of image Fig(1)(b), measure two can segment the window portions also while measure is unable to do so, rest in all images the segmentation results remains almost same for both the measures.

Table 1: MIOU score table for both similarity measure

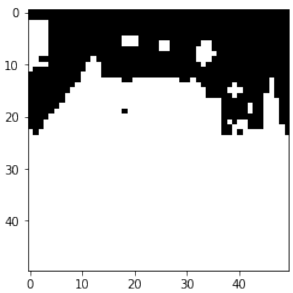
Image Name	**Measure 1 (Gaussian Kernel)**	**Measure 2 (Weibull Kernel**
Circle	0.942	0.942
Train 1	0.30	0.53
Train 2	0.37	0.49
Bird	0.59	0.23



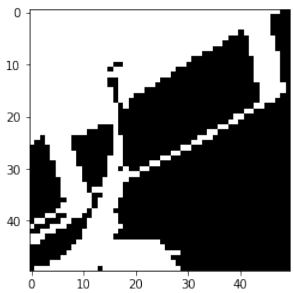
(a) Circle



(b) Train 1

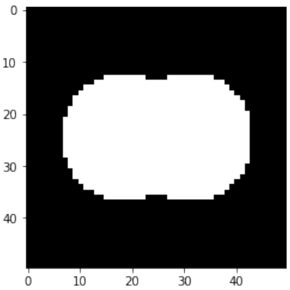


(c) Train 2

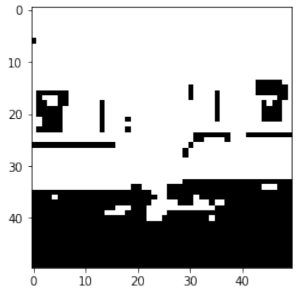


(d) Bird

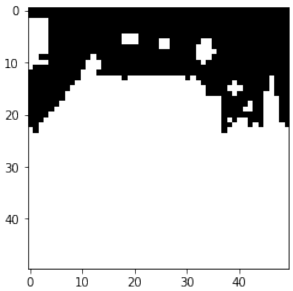
Figure 3: Segmentation using Gaussian kernel



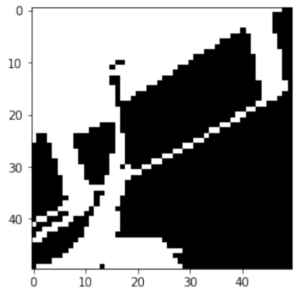
(a) Circle



(b) Train 1



(c) Train 2



(d) Bird

Figure 4: Segmentation using Weibull kernel

Question 2 Using the FCN approach (available code), segment the natural images and report the qualitative and quantitative results.

Implementation Details: I have used the pretrained ResNet101 FC model from pytorch. Only the model is made available from pytorch, the color coding is being adjusted as per PASACAL dataset. Here I have picked randomly four images from PASCAL dataset.

Results and Observations: I have computed the per pixel accuracy as well as MIOU score, the performance of pretrained model is exceptional, and very much enhanced with respect to NCut algorithm as FCN can perform semantic segmentation (NCut and FCN approach cannot be compared in terms of MIOU scores). The images used in this portion can be found on figure (5), the MIOU scores and per pixel accuracy are:

Table 2: MIOU scores and Per Pixel Accuracy for pretrained ResNet101 FCN model

Image Name	**MIOU**	**Per Pixel Accuracy (in %**
Airport	0.92	94
Plane	0.91	91.63
Computer	0.89	93.56
Train	0.90	91.93

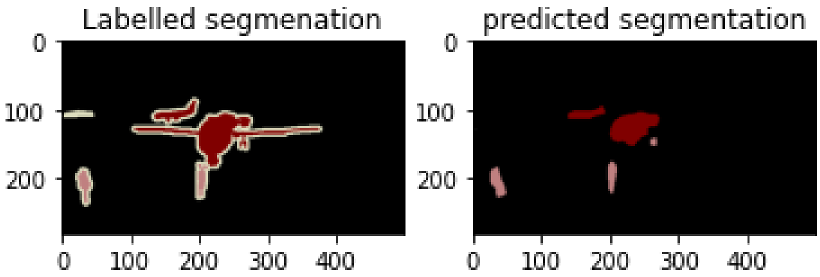
Question 3 **Choose one image classification backbone, and modify the last FC layers as taught in class (FCN paper). You can omit the skip connection part. If required, take help from the original code, but do it yourself. Segment the images as in Part 2.**

Dataset Used: I have trained my model on dataset that can be found **here**. It has 311 samples for training and 56 test images with 12 classes.

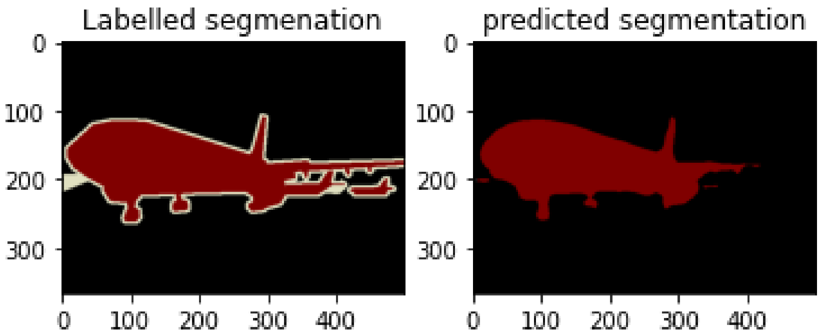
Model Specifications: I have used pretrained VGG16 model for encoder part upto block 5 pool layer. For decoder I borrowed idea taught in class, i.e using $1x1$ convolution along depth and transpose convolution for up-sampling the feature maps from encoder part. I have also added skip connection to reduce loss of spatial information. Categorical cross entropy loss is used along with Adam optimizer with learning rate of 0.001. Batch size is taken to be 16 as size of dataset is less. (Rectified the mistake of taking big batch size in previous assignment). The VGG16 encoder part has fixed weights from imagenet, the encoder is non-trainable here. Only decoder part is trained whose weights are being initialized from glorot initialize (i.e from uniform distribution). Model is being trained for 100 epochs.

Results and Observations: MIOU score of **0.42** is obtained from 56 test images. Performance is poor as compared to pre-trained model as this is trained on very limited samples. Altough, this trained network is performing better than NCut, i.e qualitatively due to semantic segmentation. The results are visually pleasent than NCut one's for obvious reasons. You can see the results in Fig(6) **NOTE**: Train and Test error with respect to epochs graph is included in the folder.

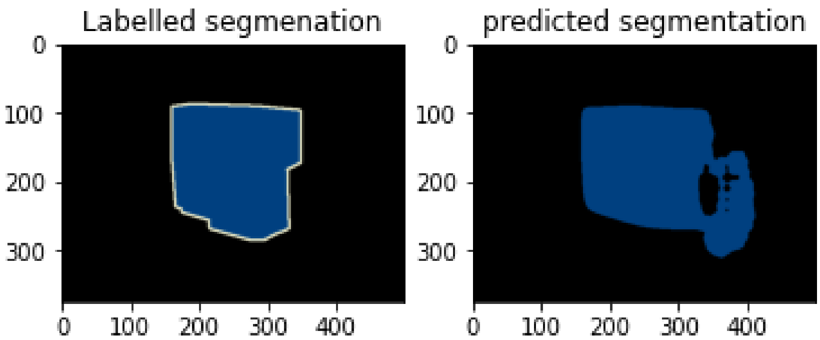
Final Conclusion and Comparison: NCut performs poor as compared to FCN approach. To make NCut give visually good results we have to play/tune with parameter like number of segments, variance for image intensity and pixel distance. Pretrained model performs the best in this assignment and the model I have trained performed better than NCut but poor as compared to pretrained. The reason is, it's being trained on very less amount of samples.



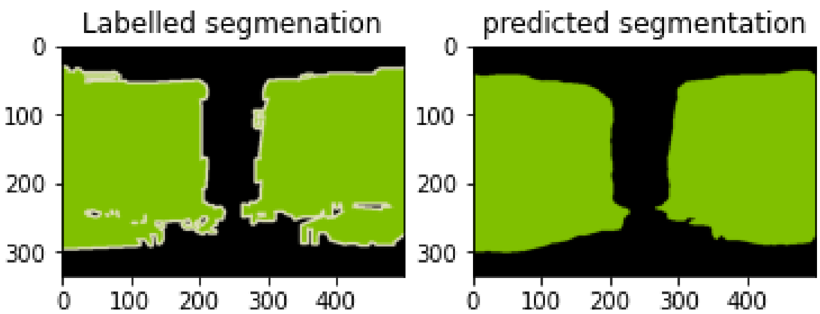
(a) Airport



(b) Plane

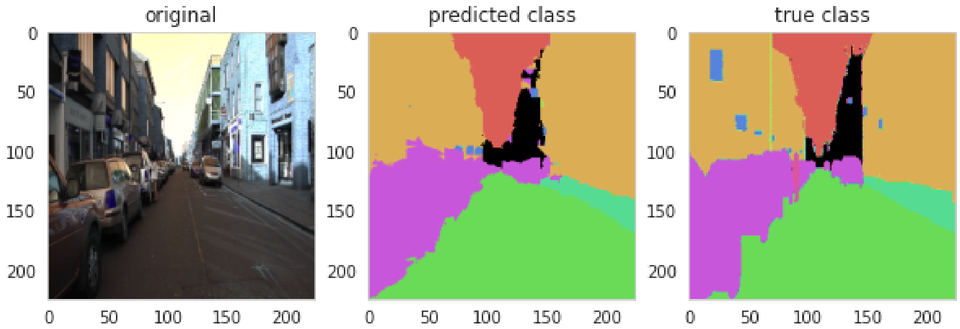


(c) Computer

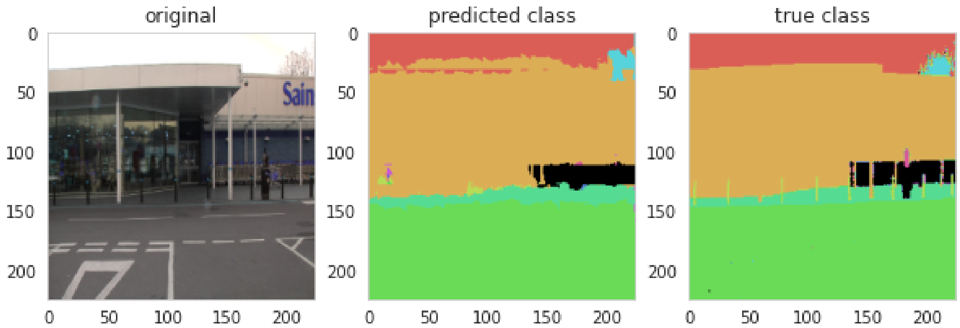


(d) Train

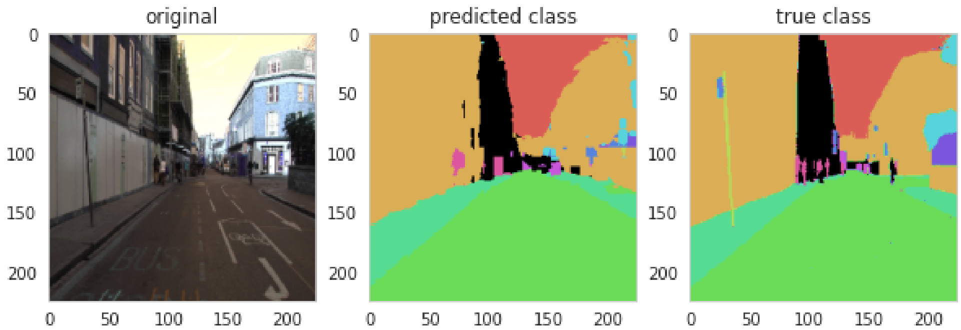
Figure 5: Segmentation using Pretrained ResNet 10



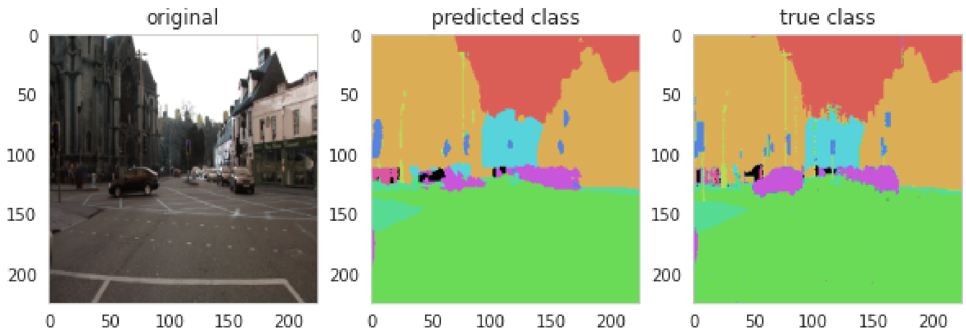
(a) Image 1



(b) Image 2



(c) Image 3



(d) Image 4

Figure 6: Segmentation using my Model (VGG16 encoder)