Implementing DeepCut

Aditya Singh
22B0056
adityas@cse.iitb.ac.in

Vishal Bysani
22B1061
vishalbysani@cse.iitb.ac.in

1 Introduction

Object segmentation is a fundamental problem in computer vision, with applications in image editing, medical imaging, and autonomous driving. Traditional methods for object segmentation require pixel-level annotations, which are expensive and time-consuming to obtain.

We have implemented DeepCut [Raj+17], a method to obtain pixelwise object segmentations given an image dataset labelled with weak annotations, in our case bounding boxes.

DeepCut falls into a class of iterative optimisation methods. It extends the approach of the well-known GrabCut [RKB04] method to include machine learning by training a neural network classifier from bounding box annotations.

2 Method

Given an image dataset $\mathcal{D} = \{(I_1, B_1), \ldots, (I_N, B_N)\}$, where I_i is the i-th image and B_i is the bounding box annotation for the object of interest, we seek to obtain pixelwise object segmentations for each image. We formulate the problem as an energy minimization task, where we seek to minimize the energy function

$$E(f) = \sum_i \psi_u(f_i) + \sum_{i<j} \psi_p(f_i, f_j), \tag{1}$$

where f_i is the label of the i-th pixel, $\psi_u(f_i)$ is the unary potential, and $\psi_p(f_i, f_j)$ is the pairwise potential.

The unary potential $\psi_u(f_i)$ is computed from a convolutional neural network that produces a distribution y_i over labels given an input image (or patch) $\mathbf{x}$ and is defined as the negative log-likelihood of this probability:

$$\psi_u(f_i) = -\log p(y_i|\mathbf{x}; \mathbf{\Theta}),$$

where $\mathbf{\Theta}$ are the parameters of the CNN. The pairwise potential is defined as

$$\psi_p(f_i, f_j) = g(f_i, f_j)[f_i \neq f_j],$$

where

$$g(f_i, f_j) = \omega_1 \exp\left(-\frac{||p_i - p_j||^2}{2\theta_\alpha^2} - \frac{||I_i - I_j||^2}{2\theta_\beta^2}\right) + \omega_2 \exp\left(-\frac{||p_i - p_j||^2}{2\theta_\gamma^2}\right). \tag{2}$$

The first term in equation 2 models the appearance and the second term in models the smoothness.

2.1 Convolutional Neural Network

For each pixel in an image, we pass a 33×33 patch centered at the pixel to a CNN. The CNN consists of two sets of convolutional layers, followed by batch normalization layers, rectified linear units (ReLU), and max-pooling layers. The architecture is shown in figure

2.2 Convolutional Neural Network Model

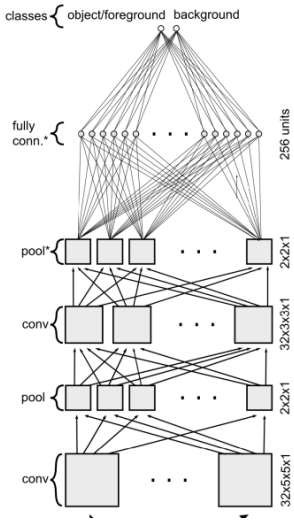

Figure 1: CNN architecture

We modified the architecture slightly by adding batch normalizaton layers, which were not present in the original paper. We also used Adam optimizer with a learning rate of 0.003 instead of Adagrad for faster learning. We use Kaiming normal [He+15] initialization for the weights of the CNN.

The loss function used to train the CNN is the cross-entropy loss between the true and predicted segmentations. Initially, every pixel in the bounding box is considered foreground, and every pixel in the halo is considered background. (The model doesn't require precise annotation of the image, only rectangular bounding boxes.)

To prevent overfitting and to speed up training, dropout regularization [Sri+14] is applied to the fully connected layers before the output layer, as well as the second pooling layer.

2.3 Conditional Random Field Regularisation

After every $N_{\text{epochs per crf}} = 15$ epochs of training the CNN, we update the pixel classes in the bounding box via inference and subsequent CRF regularisa-

ω_1, ω_2	5.0
θ_α	10.0
θ_β	20.0
θ_γ	0.5
$N_{\text{iterations}}$	5

Table 1: CRF parameters

tion. (The paper mentioned 50 epochs, but because of batchnorm layers, Adam optimizer and a smaller dataset, we found that 15 epochs were optimal.)

We made use of the python library SimpleCRF, based on [KK11]. We used the same values for the CRF parameters as in the paper [Raj+17] (for brain images) as shown in table 1.

3 Dataset

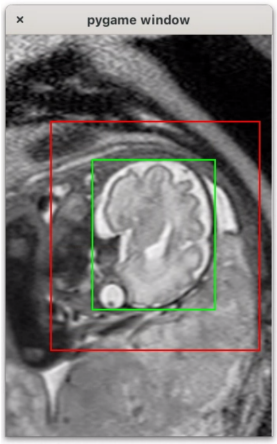

Figure 2: Bounding box (green) and halo (red) around a fetal brain.

We trained our model on fetal magnetic resonance images from [Chi]. We applied bias field correction to the images before training. The images were annotated with bounding boxes around the fetal head/lungs. We wrote a program to draw bounding boxes around the fetal head and the halo around it. The halo was drawn to include the entire fetal head and some surrounding area. The bounding boxes were used as weak annotations for training the CNN.

Our training set underwent data augmentation for better generalisation of the learned features and to prevent over-fitting to the training data as mentioned in the paper. For this purpose, we added a Gaussian-distributed intensity offset to each patch with the standard deviation 0.1 and randomly flipped the patch across the spatial dimensions to increase the variation in the training data.

Image	DSC
Image 1	0.87
Image 2	0.92
Image 3	0.87

Table 2: Dice Similarity Coefficient

4 Evaluation

The paper mentioned to evaluate the obtained segmentation against expert manual segmentations. However, since we were unable to obtain them, we manually annotated a few images and compared the obtained segmentations with them. The metric used to compare segmentations is the Dice Similarity Coefficient (DSC), which is the ratio of the intersection of the predicted and true foreground regions to the average of their areas:

$$\text{DSC} = \frac{2|A \cap B|}{|A| + |B|},$$

where A is the foreground region of our segmentation, and B is the foreground region of the expert segmentation.

The DSC is a measure of the overlap between the predicted and true segmentations, with a value of 1 indicating perfect overlap and 0 indicating no overlap.

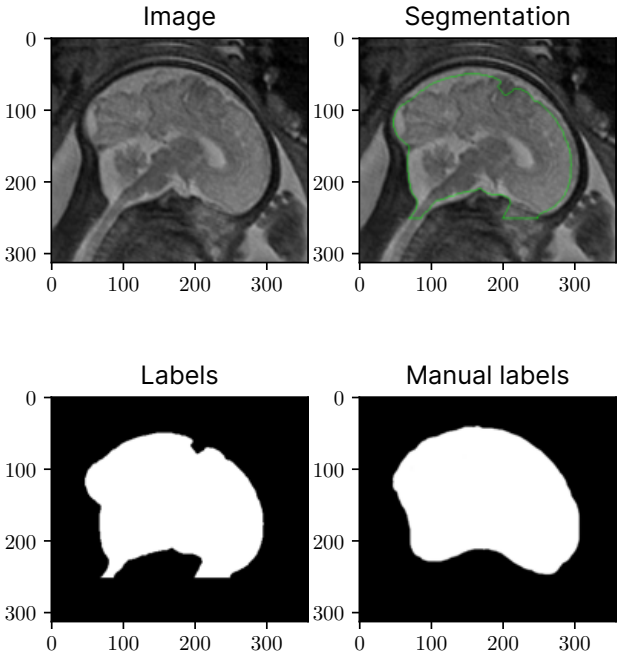

Figure 3: Image 1

5 Results

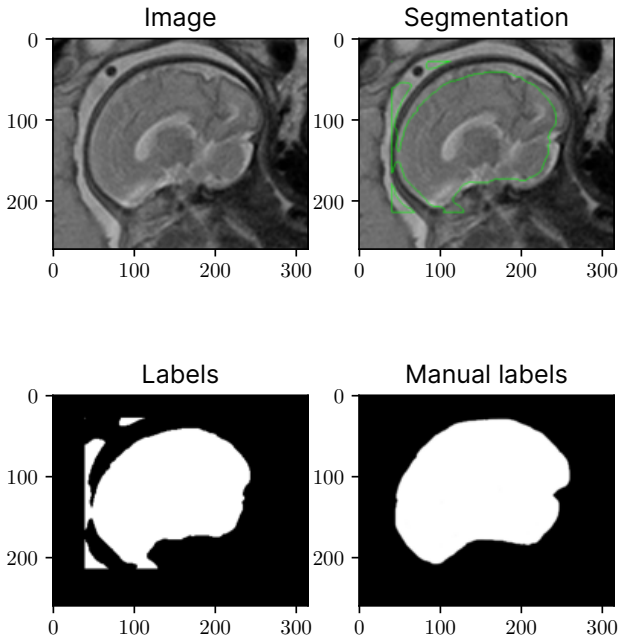

Figure 4: Image 2

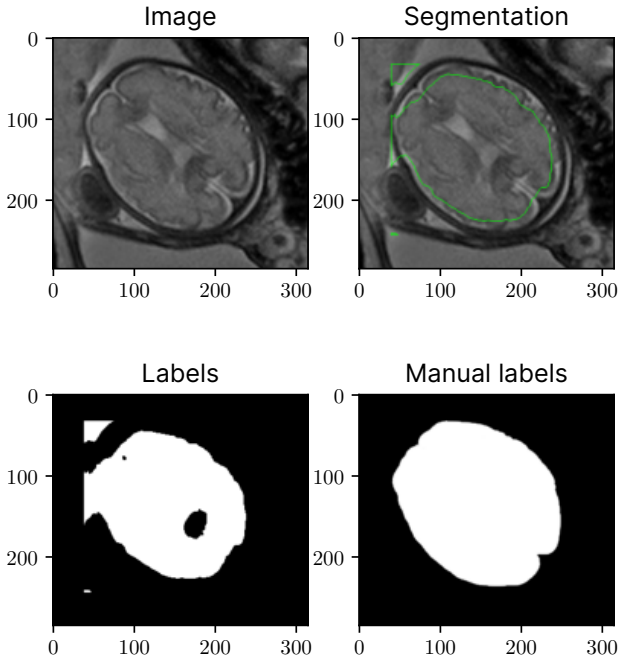

Figure 5: Image 3

We evaluated our model on 3 images. The DSC values obtained are shown in table 2. Figures 3, 4, and 5 show the original images and the obtained segmentations. The segmentations are shown in green. The foreground labels are white.

We trained our model on a single NVIDIA RTX 3060 laptop GPU with 6 GiB of memory and an Intel i7-12700H CPU with 16 GiB of system memory. Hence, we could only train on a small batch of 3 images.

6 References

[Chi] Dr. R. Chinnaiyan. *Fetal mri brain images dataset*. en. URL: https://www.kaggle.com/datasets/vijayachinns/fetail-mri-brain-images-dataset (visited on 04/30/2024).

[Raj+17] Martin Rajchl, Matthew C. H. Lee, Ozan Oktay, Konstantinos Kamnitsas, Jonathan Passerat-Palmbach, Wenjia Bai, Mellisa Damodaram, Mary A. Rutherford, Joseph V. Hajnal, Bernhard Kainz, and Daniel Rueckert. "DeepCut: Object Segmentation From Bounding Box Annotations Using Convolutional Neural Networks". In: *IEEE Transactions on Medical Imaging* 36.2 (2017), pp. 674–683. DOI: 10.1109/TMI.2016.2621185.

[He+15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015. arXiv: 1502.01852.

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *J. Mach. Learn. Res.* 15.1 (Jan. 2014), pp. 1929–1958. ISSN: 1532-4435.

[KK11] Philipp Krähenbühl and Vladlen Koltun. "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials". In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger. Vol. 24. Curran Associates, Inc., 2011. arXiv: 1210.5644.

[RKB04] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. ""GrabCut": interactive foreground extraction using iterated graph cuts". In: *ACM Trans. Graph.* 23.3 (Aug. 2004), pp. 309–314. ISSN: 0730-0301. DOI: 10.1145/1015706.1015720.