DeepCut

Based on the paper by Martin Rajchl et al[Raj+17]

Aditya Singh (22B0056), Vishal Bysani (22B1061)
adityas@cse.iitb.ac.in, vishalbysani@cse.iitb.ac.in

CS 736 (Medical Image Computing) Course Project

Contents

1. Introduction
2. Method
    Problem Formulation
    CNN
3. CRF
4. Dataset
    Annotations
5. Results

Introduction

Introduction

We implement DeepCut, a method to obtain pixelwise object segmentations given an image dataset labelled weak annotations, in our case bounding boxes. 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.

Method

Problem Formulation

- Formulate the problem as an energy minimisation problem over a densely-connected conditional random field, and
- iteratively update the training targets to obtain pixelwise object segmentations.

Energy Function

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

where $\mathbf{f}$ is the pixelwise segmentation, ψ_u is the unary potential, and ψ_p is the pairwise potential.

Problem Formulation

Energy Function

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

where $\mathbf{f}$ is the pixelwise segmentation, ψ_u is the unary potential, and ψ_p is the pairwise potential.

- The unary potential is defined as the negative log-likelihood of the pixel belonging to the object.
- The pairwise potential penalises label differences for any two pixel locations.

Energy Function

$$E(\mathbf{f}) = \sum_{i} \psi_u(f_i) + \sum_{i<j} \psi_p(f_i, f_j).$$

The unary potential term

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

where $p(y_i|\mathbf{x}; \mathbf{\Theta})$ is the probability of pixel i having label f_i.

We use a CNN to model this probability.

CNN

- The CNN is trained to predict the probability of a pixel belonging to the foreground given the input image.
- For each pixel in the image, we pass a 33×33 patch to the CNN, centered at that pixel.
- The CNN is trained using the weak annotations provided in the form of bounding boxes.
- We use the cross-entropy loss function.

Code

```
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 5)
        self.bn1 = nn.BatchNorm2d(32)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(32, 32, 3)
        self.bn2 = nn.BatchNorm2d(32)
        self.dropout = nn.Dropout(0.5)
        self.fc1 = nn.Linear(32 * 6 * 6, 256)
        self.fc2 = nn.Linear(256, 2)
        self.init_layers()
        return x
```

Code

```python
class CNN(nn.Module):
    def init_layers(self):
        nn.init.kaiming_normal_(self.conv1.weight, nonlinearity='relu')
        nn.init.kaiming_normal_(self.conv2.weight, nonlinearity='relu')
        nn.init.kaiming_normal_(self.fc1.weight, nonlinearity='relu')
        nn.init.kaiming_normal_(self.fc2.weight, nonlinearity='linear')

    def forward(self, x):
        x = self.pool(F.relu(self.bn1(self.conv1(x))))
        x = self.pool(F.relu(self.bn2(self.conv2(x))))
        x = self.dropout(x)
        x = torch.flatten(x, 1)
        x = self.fc1(x)
        x = self.dropout(x)
        x = F.relu(x)
        x = self.fc2(x)
        return x
```

CNN

We interrupt training after $N_{\text{epochs per crf}}$ epochs. We then inference from the model and use the CRF to refine the segmentation.

CRF

Energy Function

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

The pairwise potential is defined as

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

where

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

where p_i and I_i are the spatial and intensity features of pixel i.

CRF

$$g(f_i, f_j) = \omega_1 \exp\left(-\frac{||p_i - p_j||^2}{2\theta_\alpha^2} - \frac{||I_i - I_j||^2}{2\theta_\beta^2}\right) + \omega_2 \exp\left(-\frac{||p_i - p_j||^2}{2\theta_\gamma^2}\right). \quad (3)$$

- The first term models the appearance.
- The second term models the smoothness.

CRF

- The CRF minimizes the energy function using the mean field approximation.
- Instead of computing $p(\mathbf{x})$ for a labeling $\mathbf{x}$ of the image, the mean field approximation computes a distribution $q(\mathbf{x})$ that minimizes the KL-divergence $\mathbf{D}(q||p)$ among all distributions q that can be expressed as a product of independent marginals:

$$q(\mathbf{x}) = \prod_i q_i(x_i).$$

- We use the SimpleCRF library based on the paper by Krähenbühl and Koltun [KK11] to do the optimisation.

Dataset

Dataset

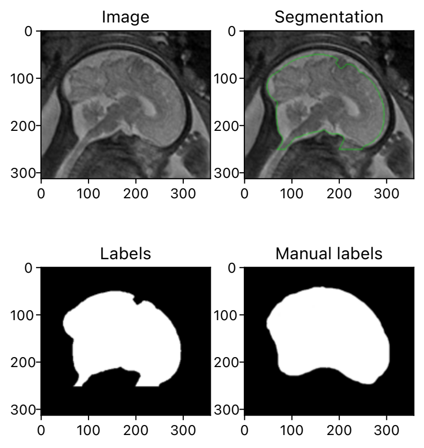
- We trained our CNN on a dataset of fetal brain MRI images [Chi].
- All images underwent bias field correction.
- We trained our model on an NVIDIA RTX 3060. So we could only train on 3 images.

Annotations

- We wrote a program to generate bounding boxes around the fetal brain.

Results

Results



... more results and DSC calculation in `report.pdf`.

References i

[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 et al. **"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 et al. **"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.

[IS15] Sergey Ioffe and Christian Szegedy. **"Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift".** In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 448–456. arXiv: 1502.03167.

[Sri+14] Nitish Srivastava et al. **"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 et al. 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.