Histopathological Image Segmentation on Breast Cancer Whole Slide Images

Different approaches

Shourya Pratap Singh - sp.singh.150604@gmail.com

Introduction

Breast cancers are the most widely reported, and subsequently the most lethal forms of cancer among women by the number of people alone. It is the 4th leading cause of cancer deaths worldwide. The diagnostic process requires histopathological analysis of the cancerous regions. This is a manual process, and is extremely tedious, with significant room for human error. To reduce the risk of inter-human variability in opinions, a machine-aided diagnostic system working as a decision support system using histopathological image segmentation is proposed. We use an unsupervised learning approach to determine intricacies via bottom-up image segmentation, and a semi-supervised learning approach to determine regions of interest from these segmented images, while navigating through annotation scarcity, and inconsistencies.

Experiments conducted so far:

1) VaDE + ViT/CNN: We are required to cluster image data without prior knowledge of class labels. The challenges lie in effectively capturing the underlying structure of the data, reducing dimensionality, and aligning the learned representations with meaningful clusters, capturing complex, non-linear relationships between image data points. A Variational Deep Embedding model is used for this clustering of image data. This is a deep generative model that combines deep neural networks of VAEs with a probabilistic framework of Gaussian Mixture Model prior in latent space. Clustering aims to group similar data points together, in an unsupervised manner. VAEs handle more complex data by learning a lower-dimensional latent representation of the data, and then a GMM prior is imposed on the latent space of the VAE, causing the learned latent representations of the VaDE to be aligned with GMM clusters.

ImageDataset is a pytorch dataset class which is used for loading and preprocessing (resize, tensor, normalize) images from a directory.

VaDE architecture:
Encoder - maps input images to latent representations.
Decoder - reconstructs the image from latent representations.
GMM parameters - learnable parameters for GMM prior for each cluster. It is initialised via k-means, uses Adam, and computed via ELBO loss.

Encoding: Image is passed through the encoder network, produces a mean vector and a log variance vector, defining a Gaussian distribution in latent space.

Input image x ⇒ encoder network ⇒ mu_z and $logvar_z$
These define a Gaussian distribution q(z|x) in latent space.

Reparameterization: A sample from the latent distribution is drawn mathematically, to allow gradient flow during sampling process while training.

q(z|x) ===> z = mu_z + eps * exp(0.5 * $logvar_z$)

[eps is a random sample from a standard normal distribution, z is a sample from the latent distribution].

Decoding: The sampled latent vector is passed through the decoder network, producing a reconstructed image.

z ⇒ decoder network ⇒ x_{recon} [reconstructed image]

The **GMM Posterior computations** calculate the probability that the sample belongs to a given cluster.

Posterior probability P(y|z) of z belonging to each cluster y=k.

Done by computing log p(z|y=k) on GMM parameters, and applying softmax to obtain probabilities.

Finally, **ELBO loss** is computed as the sum of two terms.

recon_loss: difference between x and x_{recon} using MSE.

KL divergence loss: divergence between q(z|x) and p(z) [approximate posteriors, and GMM priors].

The model is trained to minimize ELBO Loss using stochastic gradient descent. The gradients are computed with respect to the encoder, decoder, and GMM parameters, allowing the model to learn latent representations and cluster assignments simultaneously, uses Adam.

2) Shape Segmentation

A hybrid bottom-up and top-down image segmentation algorithm, combining Felzenszwalb graph-based image segmentation (BU) with template matching (TD) within a probabilistic framework.
Bottom up segmentation typically relies on low-level image features (color, texture) to group pixels into segments, while top down segmentation incorporates prior knowledge (templates, models) to guide the segmentation. We will combine both with the bottom up priors guiding the top down segmentation along with a template mask.

In the **bottom-up segmentation** process, a given image is segmented using Felzenszwalb algorithm at multiple scales, generating a hierarchy of segmentations, with finer segmentations representing more detailed smaller regions, and coarser segmentations representing larger, more abstract regions. At each scale, a directed graph where nodes represent segments is created, and the edges represent IoU similarity between segments at adjacent scales, creating a segmentation hierarchy graph. Thereafter, prior probabilities are calculated for each segment being either foreground or background. This is done by propagating information up the segmentation hierarchy. Probability of the segment being foreground is influenced by the similarity to foreground segments at coarser scales.

In the **top-down segmentation** process, a template is matched against the image. The matching scores are aggregated to produce a soft mask representing the probability of each pixel belonging to the region of interest.
Finally, the softmask and the directed graph of prior probabilities is used to compute a final segmentation, where each pixel's probability of being foreground/background is a function of both the prior probability and the soft mask value.

X_i is the random variable representing whether segment i of image X belongs to foreground or background, G is a segmentation hierarchy graph, and Y is the top-down soft mask obtained from template matching.

The algo combines $P(X_i \mid G)$ computed from the bottom-up hierarchy and the likelihood $P(Y|X_i)$ derived from template matching, to finally provide a posterior probability $P(X_i|G,Y)$. This can be simplified by approximating $P(Y|X_i)$ by exp(- λ*y) if X_i=0 else exp(- λ*(1-y)), where y is the top-down response at the pixel corresponding to segment i, and λ is a weighing factor.

3) CUTS Implementation

a) Patch Embedding: This processes images as pixel centred patches, and use a convolutional encoder without pooling to maintain spatial dimensions. Patches are extracted for each pixel, and then the sample positive patches and negative patches, and then it undergoes contrastive learning to sample positive and negative patches. Contrastive loss is then computed using cosine similarity in embedding space. Next, the embeddings are decoded back to image space, and a reconstruction loss is computed as L2 Distance. Weighted sum of Contrastive and Recon losses are minimized.
b) Diffusion: Implements the diffusion process, computes affinity matrices and diffusion operators, and performs iterative coarse-graining of embeddings; identifying persistent structures across granularities.

More at: https://github.com/KrishnaswamyLab/CUTS

4) UnSegArmaNet Implementation

Uses a ViT based feature extractor to capture global image context, creates a graph representation to understand relationships between image regions, and ARMA filters enable efficient feature aggregation across graph structure. This captures both local and global image contexts, and maintains spatial coherence in segmentation.
ViT Feature Extractor uses a DINO ViT small model with patch size 8, patches the images, and then extracts 384-D features (f) from the patches.
The image patches are represented as nodes in the graph construction process, and the edges are created wherever normalized correlation is as such: ff^T > tau. ARMA filtering is then used to implement graph convolution with skip connections, and eventually the final representations are assigned clusters. These give rise to the segmented images.

More at: https://github.com/ksgr5566/UnSeGArmaNet

References:

1) Liu, C., Amodio, M., Shen, L.L., et al. (2024). CUTS: A Deep Learning and Topological Framework for Multigranular Unsupervised Medical Image Segmentation. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2024. LNCS, vol 15008. Springer, Cham. pp. 155-165.
2) Reddy, K. S. G., Bodduluri, S., Adityaja, A. M., Shigwan, S., Kumar, N., & Mukherjee, S. (2024). UnSeGArmaNet: Unsupervised Image Segmentation using Graph Neural Networks with Convolutional ARMA Filters. Proceedings of the 35th British Machine Vision Conference (BMVC 2024). BMVA. https://papers.bmvc2024.org/0922.pdf
3) Borenstein, E., & Malik, J. (2006). Shape guided object segmentation. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 1, 969-976. https://doi.org/10.1109/CVPR.2006.111