

IFT 3710 - Team "GANg"

Data Augmentation for Cell Segmentation

Bio Samir Gbian¹, Kamen Damov¹, Simon Langlois¹, Guillaume Genois¹, and Johann Sourou¹

¹Departement of Computer Science and Operations Research

¹University of Montreal

March 8, 2025

[GitHub Code Repository](#)

1 Preprocessing

This section details the image transformation pipeline used in a microscopy image segmentation framework based on the MONAI (Medical Open Network for AI) library. The transformations are applied to both the input images and their corresponding segmentation labels to augment the dataset and improve model generalization.

Training Transformations The training pipeline incorporates both spatial and intensity transformations to create a diverse set of training samples:

$$\mathcal{T}_{train} = T_1, T_2, \dots, T_n \quad (1)$$

where each T_i represents a specific transformation operation.

Loading and Channel Transformations

$$T_1 = \text{LoadImaged}(\text{keys} = ["\text{img}", "\text{label}"]) \quad (2)$$

$$T_2 = \text{AddChannel}(\text{keys} = ["\text{label}"]) \quad (3)$$

$$T_3 = \text{AsChannelFirst}(\text{keys} = ["\text{img}"]) \quad (4)$$

These initial transformations handle the loading of images and labels, ensuring the proper channel arrangements. The images are originally in format $(H, W, 3)$ and are transformed to $(3, H, W)$, while labels are expanded from (H, W) to $(1, H, W)$.

Intensity Normalization

$$T_4 = \text{ScaleIntensity}(\text{keys} = ["\text{img}"]) \quad (5)$$

This transformation normalizes the intensity values of the input images, scaling them to a standardized range.

Spatial Transformations

$$T_5 = \text{SpatialPadd}(\text{keys} = ["\text{img}", "\text{label}"]), \quad (6)$$

$$T_6 = \text{RandSpatialCrop}(\text{keys} = ["\text{img}", "\text{label}"]), \quad (7)$$

$$T_7 = \text{RandAxisFlip}(\text{keys} = ["\text{img}", "\text{label}"]), \quad (8)$$

$$T_8 = \text{RandRotate90}(\text{keys} = ["\text{img}", "\text{label}"]), \quad (9)$$

These transformations modify the spatial properties of the images and labels through padding, cropping, flipping, and rotation operations, which introduces geometric variability into the training set.

Intensity Augmentations

$$T_9 = \text{RandGaussianNoised}(\text{keys} = ["\text{img}"]), \quad (10)$$

$$T_{10} = \text{RandAdjustContrast}(\text{keys} = ["\text{img}"]), \quad (11)$$

$$T_{11} = \text{RandGaussianSmooth}(\text{keys} = ["\text{img}"]), \quad (12)$$

$$T_{12} = \text{RandHistogramShift}(\text{keys} = ["\text{img}"]), \quad (13)$$

These transformations modify the intensity characteristics of the images through noise addition, contrast adjustment, smoothing, and histogram manipulation, simulating various imaging conditions.

Advanced Spatial Transformations

$$T_{13} = \mathbf{RandZoomd}(\text{keys} = ["\text{img}", "\text{label}"]), \quad (14)$$

This transformation randomly scales the images and labels, creating variations in object size and resolution.

Type Conversion

$$T_{14} = \mathbf{EnsureTyped}(\text{keys} = ["\text{img}", "\text{label}"]) \quad (15)$$

This final transformation ensures that the data types are compatible with the PyTorch framework.

Validation Transformations The validation pipeline is simpler and focuses on preparing the data without augmentation:

$$\mathcal{T}_{val} = T'_1, T'_2, \dots, T'_m \quad (16)$$

where each T'_j represents a validation-specific transformation operation.

$$T'_1 = \mathbf{LoadImaged}(\text{keys} = ["\text{img}", "\text{label}"]) \quad (17)$$

$$T'_2 = \mathbf{AddChanneld}(\text{keys} = ["\text{label}"]) \quad (18)$$

$$T'_3 = \mathbf{AsChannelFirstd}(\text{keys} = ["\text{img}"]), \quad (19)$$

$$T'_4 = \mathbf{ScaleIntensityd}(\text{keys} = ["\text{img}"]), \quad (20)$$

$$T'_5 = \mathbf{EnsureTyped}(\text{keys} = ["\text{img}", "\text{label}"]) \quad (21)$$

The validation transformations maintain the same preprocessing steps but exclude the augmentation operations to ensure consistent evaluation.

2 Models

2.1 U-Net

2.2 Cellpose 3.0

3 Data Augmentation

A challenge that was flagged by the other competitors was the inefficiency of using unlabeled data. We try a novel way to generate pseudo masks. We explore two ways to do so.

3.1 Pix2Pix

Generative Adversarial Networks (GANs) have been widely used for image-to-image translation tasks, and conditional GANs (cGANs) extend this concept by learning a mapping from a source domain X to a target domain Y given paired data. In this work, we leverage the Pix2Pix architecture, a cGAN designed for paired image translation, to generate pseudo-masks from raw images.

Instead of employing a traditional segmentation model, we fine-tune Pix2Pix using a corpus of labeled data consisting of (*raw image*, *segmentation mask*) pairs. The generator G in the Pix2Pix framework learns to map an input raw image $x \in X$ to a synthetic segmentation mask $\hat{y} \in Y$, while the discriminator D evaluates the realism of the generated mask by distinguishing it from real ground truth segmentation masks. This adversarial setup optimizes the generator to produce segmentation masks that closely resemble human-labeled annotations.

The objective function for Pix2Pix is formulated as:

$$\mathcal{L}_{\text{cGAN}}(G, D) = \mathbb{E}_{(x,y) \sim p_{\text{data}}(x,y)} [\log D(x, y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D(x, G(x)))], \quad (22)$$

where G attempts to minimize this loss while D aims to maximize it.

To ensure that the generated segmentation masks remain faithful to the structural information in the input image, we incorporate an additional L_1 loss:

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{(x,y) \sim p_{\text{data}}(x,y)} [\|y - G(x)\|_1]. \quad (23)$$

Thus, the full objective function is:

$$G^* = \arg \min_G \max_D \mathcal{L}_{\text{cGAN}}(G, D) + \lambda \mathcal{L}_{L_1}(G), \quad (24)$$

where λ is a weighting factor controlling the influence of the L_1 loss.

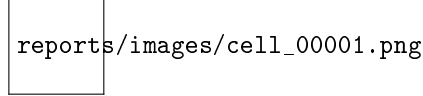
Once the Pix2Pix model is trained, we apply the generator to previously unlabeled raw images to produce synthetic segmentation masks, which we refer to as *pseudo-masks*. These pseudo-masks will then be integrated into a larger training set to improve the performance of a dedicated segmentation model. This approach allows us to expand the labeled dataset without requiring additional manual annotations, leveraging adversarial learning to refine mask generation. That said, we have issues stabilizing the training of this model, a recurring issue with GANs.

3.2 SAM-2

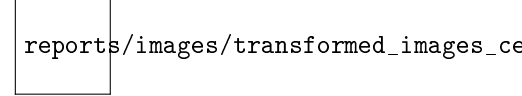
3.3 Image captionning

4 Résultat préliminaire

Voici un exemple de résultat obtenu après avoir effectué le preprocessing:



(a) Original image, 640 x 480



(b) Transformed image, 216 x 216

5 Conclusion

The comprehensive transformation pipeline described in this report facilitates robust training of deep learning models for microscopy image segmentation. The training transformations introduce diverse variations to the input data, enhancing the model's ability to generalize to unseen images, while the validation transformations ensure fair and consistent evaluation. This approach aligns with best practices in medical image analysis, where data augmentation is crucial due to the limited availability of annotated data.