CryoET Object Identification - Supervised 3D image segmentation

Program MLDM
Professor Amaury Habrard
TA Erick Gomez
erick.gary.gomez.soto@univ-st-etienne.fr

1 Introduction and Deadlines

Welcome to the Deep Learning project, a practical exploration of the concepts seen during the course applied to a **real-world challenge**. You will work on the CryoET Object Identification Kaggle competition. The project encourages experimentation with various deep learning techniques and architectures. For this project, you have to respect the following hard deadlines:

- **Wednesday, November 20**: the members of the groups must be uploaded on moodle on the dedicated resource. A group must contain between 2 and 3 members.
- **Monday, December 16**: you will have a mid-term review, you must prepare a 10/15 min presentation of your current progress and formulate the objectives for the upcoming weeks.
- **Friday, January 24**: you must submit, on the dedicated resource on moodle, both code and a complete report not exceeding 20 pages, excluding references and images. Your submission should be done by using a well-structured archive.
- **Tuesday, January 28**: you will have to present your project during a 15/20 minutes presentations followed by a series of questions. The slides of the presentation should be submitted on moodle by **January 27**.

The CryoET Object Identification challenge is currently active, and has several monetary prizes, funded by the Chan Zuckerberg Initiative. You are encouraged to participate. Additionally, a solid report, and participation in the competition is an added value for your CV.

2 Topic Presentation

2.1 Background

Cryo-electron tomography (cryoET) is an advanced imaging technique that produces 3D images, called tomograms, at near-atomic resolution. Unlike 2D images composed of pixels, these tomograms consist of voxels in a 3D space. Essentially, each tomogram can be represented as a 3D array. Each image contains objects of interest. The location and labels of this objects are located in an associated file. Using the associated file, it is possible to create a target image, where you can find the labels of the objects.

2.2 Dataset Description

The dataset comprises:

- 7 tomogram images
- Voxel spacing: 10 nm (each voxel represents a 10x10x10 nm cube)
- Associated files containing x, y, z coordinates of object centroids. If you plan to participate in the challenge, please bear in mind that if you label a voxel as being part of an object of interest (for example a apo-ferritin), the challenge accepts the labeling as correct if the labeled voxel is at a distance less than half of the radius of the particle of interest (for an apo-ferritin object, this means 3 voxels). However, you are free to choose your evaluation metric. For more information see the section Report.
- Example: "TS_69_2" experiment contains 37 ribosomes, or in other words, there are 37 voxels labelled as ribosomes in the target image, they represent the centroids of the ribosomes.

2.3 Project Objective

The goal is to identify the locations of five types of objects within the tomograms:

- Ribosome (radius: 150 nm, or 15 voxels)
- Virus-like particles (radius: 135 nm, or 13.5 voxels)
- Apo-ferritin (radius: 60 nm, or 6 voxels)
- Thyroglobulin (radius: 130 nm, or 13 voxels)
- β-galactosidase (radius: 90 nm, or 9 voxels)

Object locations are expressed as centroid x, y, z coordinates.

2.4 Existing Approaches

The Kaggle competition's code section currently features three Python scripts:

- BlobDetector: Uses image preprocessing and the watershed algorithm (non-Deep Learning approach)
- DeepFindET_Train and DeepFindET_Inference: Implement a 3D U-NET segmentation model

For a quick introduction to U-NET architecture, refer to this tutorial.

2.5 Dataset Access

- Use only the denoised version of the images
- Additional synthetic data is available and may (or may not) improve results
- Due to potential space limitations, a shared data path is provided: $/home\_expes/collections/kaggle-competition-CryoET/$. You are welcomed to use this path in your scripts in case you employ the cluster.

2.6 Grading

The following are soft guidelines. In general, the grade is proportional to the quality of the final report, the defense, and the implementation of different learning techniques.

- 7/20 Explain the problem. Implement the architecture in the example.
- 10/20 Idem as above. Also modify the model, create visualizations of the data and results. Each member of the group should have focused on a particular contribution.
- 12/20 Idem as above. Also propose another architecture and compare.
- 14/20 Idem as above. Also a implement another architecture and image preprocessing and compare.
- 16/20 Idem as above. Deliver a report and defense of high quality. The code should be extremely clear and well commented.
- Extra point(s) if your team participates in the Kaggle competition.
- Extra point(s) if you compare the performance of your architectures (or do transfer learning) using also synthetic data

2.7 Report

The report must contain, not exclusively, the following sections.

1. Introduction. The goal of this section is be capable to explain in simple terms and briefly the nature of the data and the problem you try to solve. Is the problem supervised or unsupervised? Where do the images come from? Why is it important to identify this objects? The goal of this section is be capable to explain in simple terms the nature of the data and the problem you try to solve.

2. Methodology. Here you should describe the dataset (source, size, preprocessing steps if any), the architecture(s) of the model(s). Justify why you chose to implement a particular technique. Next, it is important to have a subsection on the training process: hyperparameters, optimization algorithm, scheduling, etc.

3. Experimental evaluation. Your setup should be precisely made. You are free to choose your evaluation metric, or use the metric suggested by Kaggle and used to it evaluate prediction.

   The Kaggle challenge employs a specialized F1 measure to evaluate model performance for prediction (not training). Specifically, it uses an F1 score with $\beta = 4$, which places greater emphasis on recall over precision. This choice reflects the challenge's prioritization of identifying as many relevant objects as possible, even at the cost of some false positives (prioritize recall). The evaluation uses micro-averaging, meaning that the F1 score is calculated globally across all instances (i.e. 3D images), rather than separately for each class and then averaged. Additionally, the challenge incorporates different weightings for various object types, reflecting their relative importance or difficulty in identification. For more details, please visit the web site of the challenge. It's crucial to note that this evaluation metric is specifically designed to assess the model's performance during prediction (inference) phase. During the training phase, different loss functions that are amenable to differentiation may be used. For example, the Tversky or Dice loss functions are chosen for their mathematical properties that facilitate effective gradient descent, while the F1 score is selected for its interpretability and alignment with the task's goals during evaluation.

   In this section is important explain the different evaluation measures you use. If you chose to use the F1 score suggested by the competition, keep in mind that during prediction (inference) a voxel classified as being part of an object is a true classification if the distance between the voxel and the centroid of the object is less than half the radius of the object.

   An option that could help you during training and you could keep for prediction is to represent each object as a sphere in the target image, with the radius corresponding to the object's size (e.g., ribosomes as 15-voxel radius spheres). In other words, instead of having just one voxel to represent the centroid of an object, you create a solid sphere around the centroid. In this manner your training could be easier.

4. Comparison of Architectures. This part can contain three subsections: interpretation of results, object label analysis, and failure analysis.

   Interpret Results. Compare the performance metrics (such as F1 score, precision, and recall) of the various architectures you implemented. Discuss which model performed best overall and why. Highlight the strengths and weaknesses of each architecture. For example, did one architecture excel in detecting certain types of objects while another struggled?

   Object Type Analysis. Analyze whether your models are better at recalling specific types of objects. Are there particular classes where the models perform exceptionally well or poorly? Discuss potential reasons for these differences in performance, such as object radius, frequency in the dataset, or distinguishing features.

   Failure Analysis. Describe situations where your models failed to perform adequately. Provide specific examples of misclassifications or missed detections. This is paramount to elucidate possible directions to improve your architecture.

5. Conclusion: Present a summary of your key findings.

6. Add a section to describe the unique contributions of each team member to the project and indicate the workload percentage between the members of the group. Clearly outline the specific responsibilities, tasks, and innovations brought forward by each team member. Acknowledge the collaborative efforts and synergies that contributed to the project's overall success. You can also mention positive and negative points raised during the project and comment on the expected and real planning.

The deadlines for this project are strict. We encourage you to plan your work carefully and adhere to a gradual progression throughout the project. It is possible that it could take several days only trying to read and visualize the data.