Cryo-electron Tomography Object Identification Supervised 3D image segmentation and 2D object detection

Cesar Anasco [*1] **DoPhan BaoKhang** [*1]

Abstract

The "CZII - CryoET Object Identification" competition on Kaggle challenges participants to identify small biological structures within large 3D volumes obtained through cryo-electron tomography (Cryo-ET). This report shows the solution implemented with a 3D-Unet and 2D object detection model for multi-class segmentation, the results obtained show that both approaches achieve accuracy in more than 50% of cases, with some specific classifications being better than others.

1. Introduction

The CryoET Object Identification challenge is funded by the Chan Zuckerberg Initiative and its primary objective is to acquire more knowledge about protein complexes for cellular function, which are essential for disease treatments. The available data obtained from tomographs are often available in a standardized format, and the analysis of this specific information is challenging when identifying the types of protein complexes within these images.

Cryo-electron tomography opens the door to the study of the structure of unique objects, such as cell structures and even entire cells (Stewart, 2017). To do this, multiple images of the sample are taken at different inclinations within the microscope (generally from -70º to 70º), which are subsequently processed using specialized programs to reconstruct its three-dimensional structure, as seen in Appendix A. The available dataset provided in the competition contains already classified and denoised images of tomographs, the classification includes six particle types with varying prediction difficulty: apo-ferritin (easy), beta-amylase (not scored, impossible), beta-galactosidase (hard), ribosome (easy), thyroglobulin (hard), and virus-like-particle (easy), with beta-amylase excluded from scoring due to its evaluation challenges.

[*]Equal contribution [1]Université Jean Monnet. Correspondence to: Cesar Anasco <cesar.anasco@etu.univ-st-etienne.fr>, DoPhan BaoKhang <first2.last2@www.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

2. Methodology

2.1. Dataset

The dataset consists of 7 cryo-electron tomography (cryoET) images, represented as 3D tomograms where each voxel corresponds to a 10x10x10 nm cube, as seen in Appendix B. Each tomogram contains various objects of interest, whose locations are provided as centroid coordinates in associated files. Objects include ribosomes, virus-like particles, apo-ferritin, thyroglobulin, and B-galactosidase, with radius ranging from 6 to 15 voxels. The challenge allows a voxel-level labeling to be considered correct if it falls within half the particle's radius from the actual centroid. There are associated files to each tomogram containing x, y, z coordinates of object centroids.

Synthetic data has been used to train models to detect these objects. This data is generated with realistic characteristics mimicking the tomograms, serving as a proxy for real-world samples, especially when annotated real tomograms are limited.

For the second architecture implemented, the preparation of the datasets for training converts 3D volumetric data into 2D images slices, this reduces memory requirements and address data scarcity. Key steps in this process include normalizing the data, creating image slices, generating YOLO-compatible annotations, and organizing datasets into structured folders for training and validation.

2.2. Architectures Implemented

There are numerous architectures, methods and approaches that have proven to be especially effective in certain object detection tasks and for extracting features of tomograms. Among these, the YOLO (You Only Look Once) network (Diwan et al., 2023) and 3D U-NET (Agrawal et al., 2022) stand out.

2.2.1. 3D U-NET

3D U-Net is a convolutional neural network (CNN) architecture designed specifically for image segmentation tasks, where the goal is to classify each pixel (or voxel in 3D cases) in the input image. It is particularly well-suited for biomedical image analysis, making it ideal for the cryo-electron tomography (cryoET) dataset.
Key features of the model 3D U-Net are:

- Encoder-Decoder Structure:
  1. Encoder: responsible for capturing contextual information by downsampling the input image through convolutional, max-pooling layers and extracting high-level features.
  2. Decoder: responsible for reconstructing the spatial details by upsampling the features back to the input resolution and producing a dense segmentation map.
- Skip Connections: while encoding, the model also send the outputs to the corresponding layers and the Decoder help recover fine-grained spatial details lost during downsampling. These connections concatenate feature maps from the Encoder with those in the Decoder, enhancing localization accuracy.
- 3D Adaptation: For the CryoET dataset, the 2D U-Net is extended to a 3D version, where 3D convolutions and pooling operations are applied, enabling the model to process volumetric data and segment objects in 3D space effectively.

2.2.2. YOLO

This method incorporates a real-time object detection stage, which uses a convolutional neural network to divide a 2D image into regions and predict the coordinates and probabilities of existence of the objects in each region. YOLO has the advantage of being fast, accurate and robust against different lighting conditions, size and shape of the objects.

Several studies have applied YOLO in detection tasks from medical images: such as mammography (Al-Masni et al., 2018), the study of melanoma (Nie et al., 2019) and dental diseases (Sonavane & Kohar, 2022), ,achieving accuracies and sensitivities above 90% in laboratory simulations and studies with patients, for internal validation data. These results may be an indicator of the effectiveness of the method in detecting abnormal objects in biomedical tasks, as well as its potential to become a novel approach, capable of performing disease detection and classification with good performance in clinical routine, which could also have relevant implications for this specific competition.

The Python programming language and PyTorch machine learning frameworks were used. Additionally, tools and libraries such as YOLOv5 from Ultralytics were employed to facilitate model training and object detection. The YOLO architecture was implemented using a pre-trained model (YOLO11), which incorporates recent advancements in object detection and data augmentation. Data pre-processing and augmentation included techniques like rotation, shear, flipping, and mix-up during training. The development and training process used a machine equipped with an NVIDIA L4 GPU (22.5 GB GDDR6), providing the computational power necessary to efficiently train the model + 235 GB HDD storage.

The latest YOLO11 model architecture, as shown in Appendix F, is composed by three main parts:

- Backbone: Is the deep learning architecture that acts as a feature extractor.
- Neck: Combines the features acquired from the various layers of the backbone model.
- Head: Predicts the classes and bounding box regions which is the final output produced by the object detection model.

2.3. Training Process

2.3.1. Transfer Learning with Synthetic Data

Before training the model with real-world data, we opted for a transfer learning approach by pre-training the model on synthetic data. Synthetic data often provides a controlled environment where particle features, distributions, and annotations are more reliable and consistent than in real data. Pre-training allows the model to learn general features and patterns that are transferable to real-world data, such as recognizing particle shapes and boundaries.

2.3.2. Pre-trained 3D U-NET

During training, the validation metric used in this model is Dice Metric, which is commonly used in segmentation tasks to evaluate the overlap between the predicted segmentation and the ground truth. It is not the same as ”accuracy” in a traditional classification sense but is instead a measure of how well the predicted and true segmentation align.
The Dice score ranges from 0 to 1:

- 1 indicates perfect overlap (the prediction is exactly the same as the ground truth).
- 0 indicates no overlap.

It is computed as:

$$Dice\ Score = \frac{2|A \cap B|}{|A| + |B|} \tag{1}$$

Where:

- A is the predicted segmentation
- B is the ground truth segmentation.

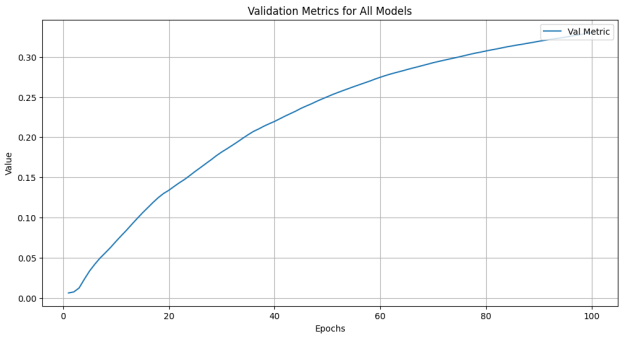

Figure 1. Validation Score Performance during training phase

The model performs well while training with the validation score increasing.
And as for the loss function used in 3D U-Net model is the Tversky Loss, which is particularly suited for imbalanced segmentation tasks, especially when one class significantly dominates the others.
The Tversky Loss is a generalization of the Dice Loss and is defined as:

$$Tversky\ Index = \frac{TP}{TP + \alpha.FP + \beta.FN} \tag{2}$$

$$\alpha + \beta = 1 \tag{3}$$

$$Tversky\ Loss = 1 - Tversky\ Index \tag{4}$$

Where:

- TP: True Positives.
- FP: False Positives.
- FN: False Negatives.
- α, β: Weighting parameters that balance the importance of false positives and false negatives.

In case of equation (2) if $\alpha = \beta = 0.5$, this loss is equivalent to the Dice Loss.

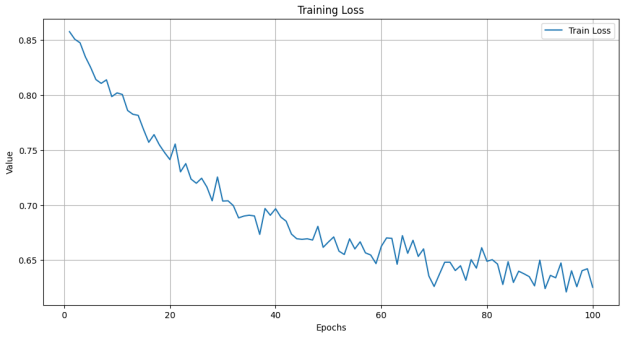

Figure 2. Loss Performance during training phase

The model's loss initially decreases well at first but then struggles to improves from around epoch 50.

2.3.3. YOLO

The training configuration includes the following parameters and optimization techniques:

- Epochs: The model is trained for 100 full passes through the dataset.
- Warm-up Epochs: Gradual increase in the learning rate over the first 10 epochs.
- Batch Size: The number of samples processed in one training step is 32.
- Image Size: Input images are resized to 640 x 640 pixels.
- Data Augmentation:
  - Rotation: Up to ±45 degrees.
  - Shear: Up to 5 degrees.
  - Horizontal and Vertical Flipping: Probabilities of 0.5 for both.
  - Mixup: A data augmentation technique that blends two training images.
  - Copy-Paste: Augmentation by combining regions from different images.
- Optimizer: AdamW
- Seed: 8620 – Used for reproducibility.
- Initial Learning Rate: 0.0003

Losses in Figure 3 are divided in three terms that define the overall loss function used in object detection models like YOLO11. The total loss function for object detection is given by:

$$\text{Loss} = \alpha \cdot \text{DFL Loss} + \beta \cdot \text{CLS Loss} + \gamma \cdot \text{BOX Loss} \quad (5)$$

Where:

- α: Weight for the Distribution Focal Loss (DFL Loss),
- β: Weight for the Classification Loss (CLS Loss),
- γ: Weight for the Bounding Box Loss (BOX Loss).

The total loss can be calculated programmatically as:

$$\text{Total Loss} = \alpha \cdot dfl + \beta \cdot cls + \gamma \cdot box \quad (6)$$

3. Experimental evaluation setup

3.1. 3D U-Net

During evaluation phase on the test dataset, we follow the official metric score of the Kaggle competition which focuses on precision, recall, and F-beta score.

1. Precision: Measures the proportion of correctly predicted objects (hits) among all predicted objects.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

2. Recall: Measures the proportion of correctly predicted objects among all ground-truth objects.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

3. F-beta Score: A weighted harmonic mean of precision and recall, emphasizing recall (with β=4):

$$F_\beta = \frac{(1+\beta^2).Precision.Recall}{(\beta^2.Precision) + Recall}$$

In this case, β=4 means recall is weighted 16 times more than precision.

In the context of this challenge, a particle is considered a "true" positive if it lies within 0.5 times the particle's radius of the ground truth particle, this tolerance helps account for some variability in particle locations, allowing small shifts while still counting as a correct prediction. And the particles are divided into 3 types and weighted differently:

1. Easy Particles (ribosome, virus-like particles, apoferritin) are assigned a weight of 1.

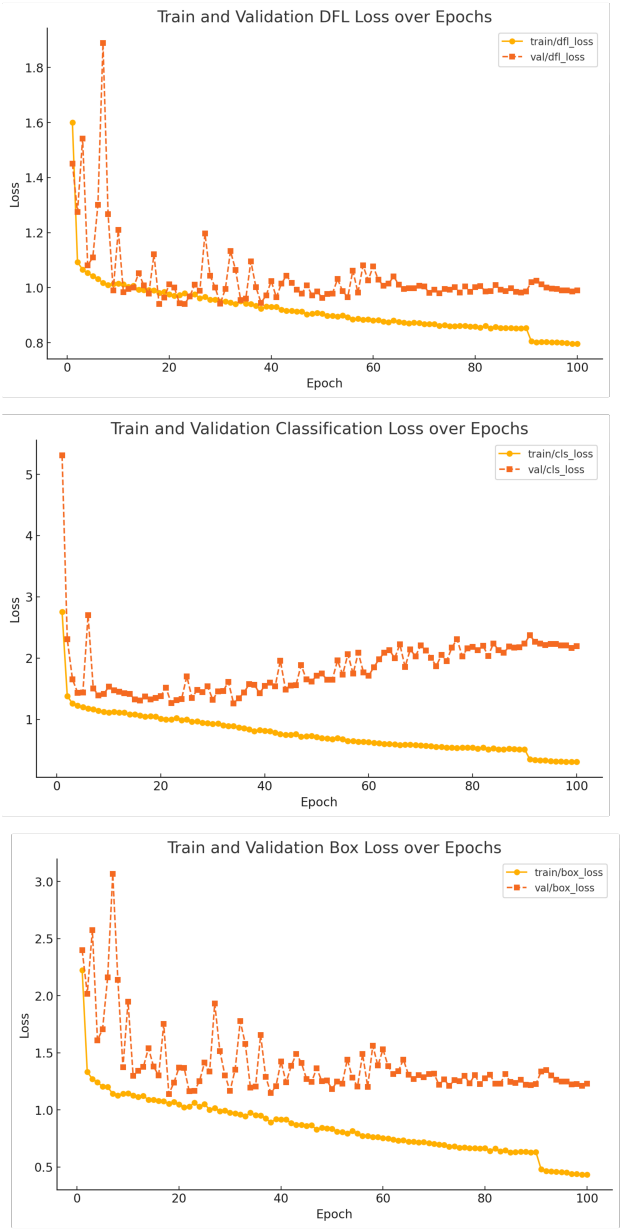

Figure 3. YOLO Training Losses Result

2. Hard Particles (thyroglobulin and β-galactosidase) are assigned a weight of 2. This weighting scheme reflects the relative difficulty of detecting each particle type, with harder particles having more influence on the final score. The hard particles are particularly prioritized, making recall critical for them.

3. Impossible Particles: Beta-amylase particles are included in the training data but do not contribute to the score, as they have a weight of 0 in the scoring mechanism. Even if predicted, they do not affect the final

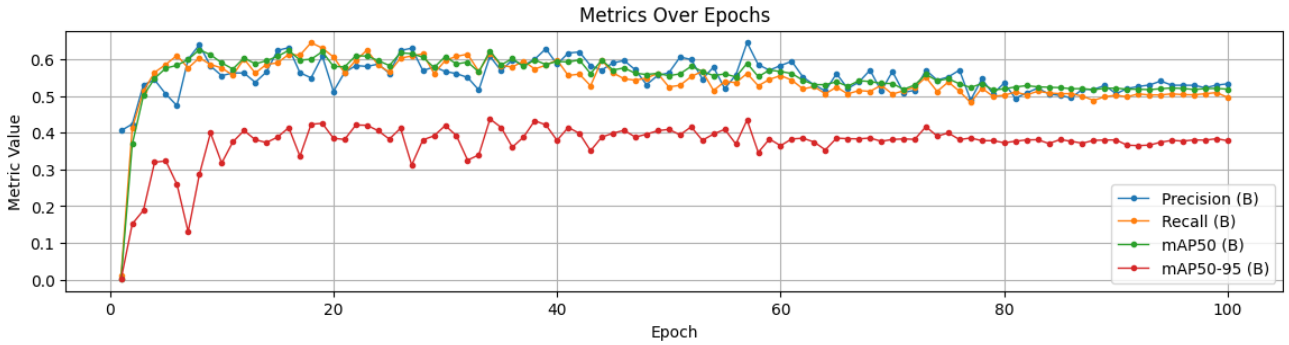

Figure 4. Evaluation Metric for YOLO Model Training

evaluation.

The final F-beta score is computed by summing the per-particle scores, applying the weights to each particle type, and normalizing by the total weight:

$$Final\ lb\_score = \frac{\sum(F_4(particle).weight(particle))}{\sum weight(all\ particles)} \tag{7}$$

This gives a Final lb_score that reflects the model's ability to identify particles correctly, with emphasis on the harder particles and recall.

3.2. YOLO

Figure 4 graph shows the evaluation metrics on the YOLO11 model over 100 training epochs. Each line tendency and definition are described below:

- **Precision (Blue Line):** measures how many detected objects are classified correctly. The precision starts low and starts increasing rapidly in the first 10 epochs, as the training configurations makes learning rate increase after 10 epochs. After this, precision stabilizes above **0.6**.
- **Recall (Orange Line):** measures how many of the ground truth objects are correctly detected. It has the same growing tendency as precision.
- **mAP50 (Green Line):** Mean Average Precision at 50% IoU (Intersection over Union). This metrics shows the accuracy for both classification and localization when the IoU threshold is 50%. In the graph, this metric increases significantly in the first 10 epochs and then stabilizes around **0.55–0.6**.
- **mAP50-95 (Red Line):** Mean Average Precision across IoU thresholds from 50% to 95%, in increments of 5%. This metric evaluates performance across levels of detected objects overlapping. This metric shows slower growth compared to the metrics tendencies, stabilizing around **0.4–0.45**.

This evaluation show that the model achieves reliable performance in both classification (Precision, Recall) and localization (mAP metrics) after 100 training epochs.

4. Results and Discussion

As a result, the 3D-Net model pre-trained with the synthetic data outperforms the baseline model in terms of final F-beta score ($0.524 > 0.339$). YOLO11 model was trained with the provided competition dataset and outperforms the 3D implementation. Both approaches were published as submission for the kaggle competition.

4.1. Comparison of Architectures

Both architectures implemented, 3D U-NET and YOLO11, are based on convolutional neural networks (CNNs) as their backbone. As their names says, 3D U-NET uses 3D convolutional layers to process volumetric datasets, while YOLO processes object detection in 2D images.

4.2. Interpretation of results

Table 1 shows the performance that both models achieved after submission to Kaggle. Submissions are evaluated by calculating the F-beta metric with a beta value of 4. In this case YOLO11 training got the best score, this result can be justified with the fact that YOLO models have faster inference and training time due to their 2D design. The can achieve high accuracy in detecting object centers when working with individual slices. In contrast, 3D U-NET achieved a 0.33 score but at the cost og higher computational costs. 3D can better capture volumetric context, and with the use of synthetic data we were able to achieve a better score, but we were far behind YOLO11.

Model	Kaggle Score
U-NET Baseline	0.339
Pre-trained 3D U-NET	0.524
Pre-trained YOLO 11	0.625

Table 1. Results Comparison of Submissions Scores

4.3. Object label analysis

4.3.1. 3D U-NET

From Table 3, we can observe the following attributes of the model prediction on all the particles:

1. "Easy" particles:
   - Virus-like-particle achieves the highest F-beta score (0.826), indicating effective identification.
   - Apo-ferritin had the lowest F-beta score (0.378), suffer from low precision.
   - Ribosome also achieves a high F-beta score (0.703), after only Virus-like-particle.

   Overall the "easy" particles are truly easier to detect, except for Apo-ferritin.
2. "Hard" particles:
   - Thyroglobulin, despite being harder, shows strong recall, contributing positively to the overall evaluation.
   - Beta-galactosidase suffer from low precision, which reduces its F-beta scores.
3. "Impossible" particle:
   Although being listed as impossible, the model are able the detect the beta-amylase particle not so bad with F-beta score of 0.515, even better than apo-ferritin and beta-galactosidase.

We also recognize that particles with high recall often have lower precision, suggesting the model prioritizes identifying true positives but struggles with false positives and "Hard" particles tend to have more challenges, particularly in precision.

4.3.2. YOLO11

YOLO11 results can be seen in Appendix G, where each cell contains the proportion of predictions for that combination of true and predicted classes. The most important observations are:

- Ribosome: 78% of the ribosome instances were correctly classified. This is a relatively strong performance for this class.
- apo-ferritin: 62% of predictions are correct, but there is some misclassification (38% in other classes).
- virus-like-particle: Performs the best, with 90% correct predictions.

4.4. Failure Analysis

4.4.1. 3D U-NET

The model occasionally fails to identify true particles, leading to high recall penalties. In case of the "easy" particle apo-ferritin, the model detected only 51 out of 139 true particles, resulting in a recall of 36.69%. This suggests that the model struggles to generalize apo-ferritin features, possibly due to overlapping characteristics with other particles or insufficient training examples in the synthetic dataset which leads to a high miss rate reducing the F-beta score and the final lb_score of the model.

4.4.2. YOLO

For YOLO11 case the particle types beta-amylase and beta galactosidase have significant miss-classification rates, with proportions spread across other classes. For the thyroglobulin class, while 47% of the predictions are correct, it is also misclassified as "ribosome" and "virus-like particle." Also, the background of the images are misclassified, which might indicates a challenge for the model to differentiate background noise from actual particle types.

5. Conclusion and Future Work

The research focuses on particle detection and classification in Cryo-ET dataset in which we use 2 CNN-based models YOLO11 and 3D U-Net and as a result, YOLO11 outperforms 3D U-Net in most particle type classifications in terms of F-beta score. It is important to mention that the types of proteins with the highest accuracies are due to the radius around the centroid being the largest in ribosome, thyroglobulin, and virus-like particles.
In the case of YOLO11, we can see that it struggles with objects that span multiple slices; however, 3D U-Net captures volumetric context better, especially for irregularly shaped objects. This can be seen in the improvement of the score after the Kaggle submission baseline. YOLO11 could benefit still from adding of space partitioning data structures to better associate detected objects with their true 3D positions based on centroid data.
In conclusion, the choice between YOLO11 and 3D U-Net would be in favor of YOLO11 who have proved its potential in object detecting task like this but still should depend on the specific characteristics of the dataset and the task's priorities. Future work could explore hybrid approaches that combine YOLO11's speed and precision with the volumetric insights of 3D U-Net, aiming for a more comprehensive

solution to cryo-ET particle detection challenges.

6. Percentage of Involvement

Name	Percentage
DO PHAN Bao Khang	50%
Cesar Anasco	50%

Table 2. Percentage of Involvement of Each Group Member

Both team members contributed equally to the research and analysis of the data and problem. DO PHAN worked on implementing, studying the structure of 3D U-Net model and applied the proposed synthetic data in order to try to enhance the performance of the model. Cesar Anasco worked on implementing and debugging the YOLO11 baseline, which was designed for object detection within 2D slices. Additionally, both members cooperated in addressing issues during training and adding synthetic data to the baselines.

References

Agrawal, P., Katal, N., and Hooda, N. Segmentation and classification of brain tumor using 3d-unet deep neural networks. *International Journal of Cognitive Computing in Engineering*, 3:199–210, 2022.

Al-Masni, M. A., Al-Antari, M. A., Park, J.-M., Gi, G., Kim, T.-Y., Rivera, P., Valarezo, E., Choi, M.-T., Han, S.-M., and Kim, T.-S. Simultaneous detection and classification of breast masses in digital mammograms via a deep learning yolo-based cad system. *Computer methods and programs in biomedicine*, 157:85–94, 2018.

Diwan, T., Anirudh, G., and Tembhurne, J. V. Object detection using yolo: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, 82(6):9243–9275, 2023.

Nie, Y., Sommella, P., O'Nils, M., Liguori, C., and Lundgren, J. Automatic detection of melanoma with yolo deep convolutional neural networks. In *2019 E-Health and Bioengineering Conference (EHB)*, pp. 1–4. IEEE, 2019.

Sonavane, A. and Kohar, R. Dental cavity detection using yolo. In *Proceedings of Data Analytics and Management: ICDAM 2021, Volume 2*, pp. 141–152. Springer, 2022.

Stewart, P. L. Cryo-electron microscopy and cryo-electron tomography of nanoparticles. *Wiley Interdisciplinary Reviews: Nanomedicine and Nanobiotechnology*, 9(2): e1417, 2017.

A. Cryo-electron tomography (cryoET)

This Figure 5 illustrates the process of electron tomography, a technique for 3D reconstruction of a sample.

- Top illustration: An electron beam passes through a sample that is tilted from -60 ° to +60 °, capturing two-dimensional projection images at various angles.
- Bottom illustration: The tilt series images are combined using computational methods to reconstruct a detailed three-dimensional model of the sample, revealing its internal structure.

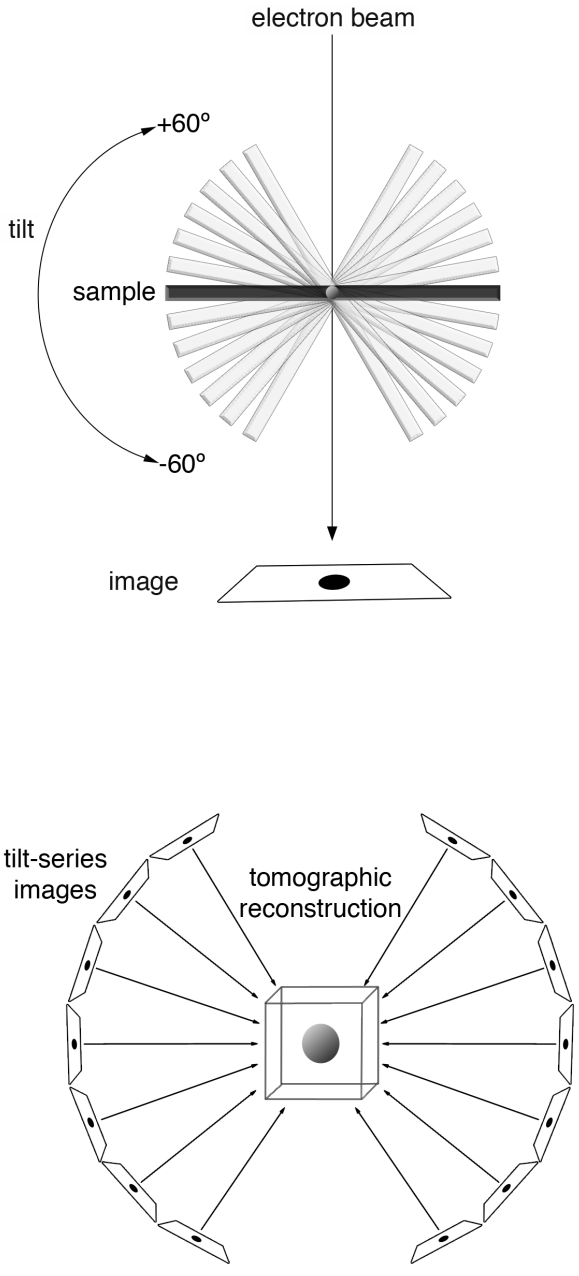

Figure 5. Construction of a Tomogram

B. Voxel Segmentation

A voxel (volumetric pixel) represents a single unit of 3D space in the cryo-electron tomography (CryoET) data, it defines the smallest measurable cube in the tomogram, capturing intensity or density information about the structure being imaged. The voxel creates segmentation labels for tomograms based on predefined properties of particles.

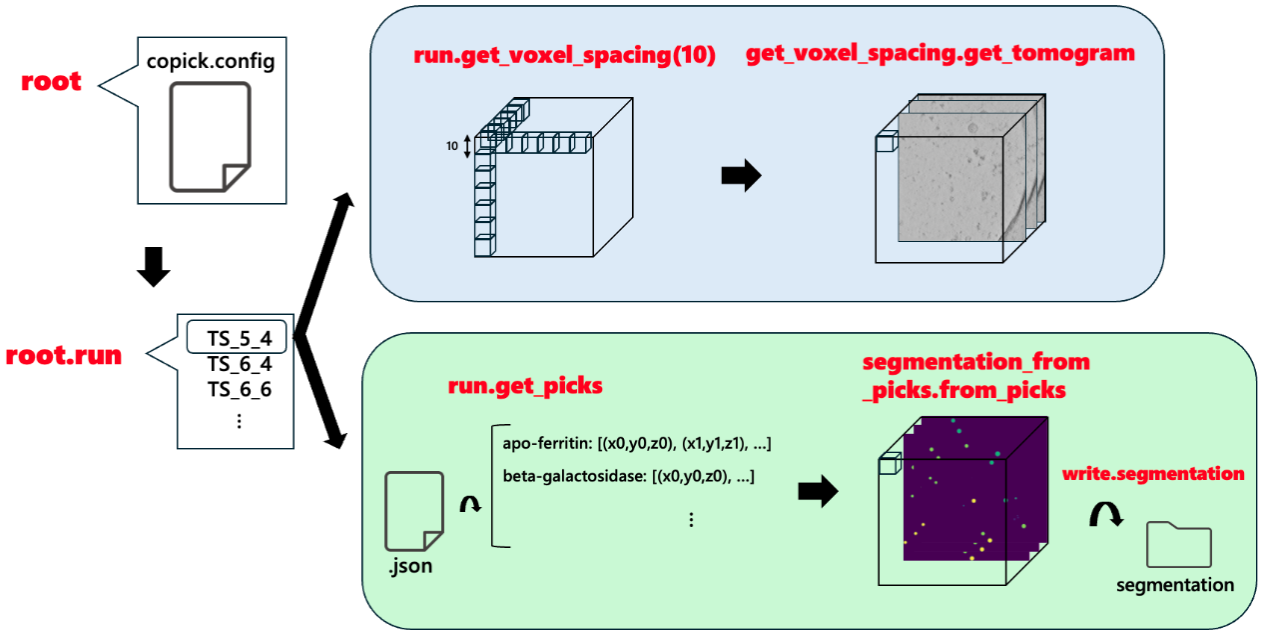

Figure 6. Voxel Segmentation in Tomograms

C. 2D Particles Detection

The following images show the prediction made by the model on the validation tomogram TS_5_4. The numbers next to each particle name in the prediction image represent the confidence score assigned by the YOLO model to each detected object. For example: A label like thyroglobulin 0.83 means that the model has 83% confidence that the detected object is of type "thyroglobulin."

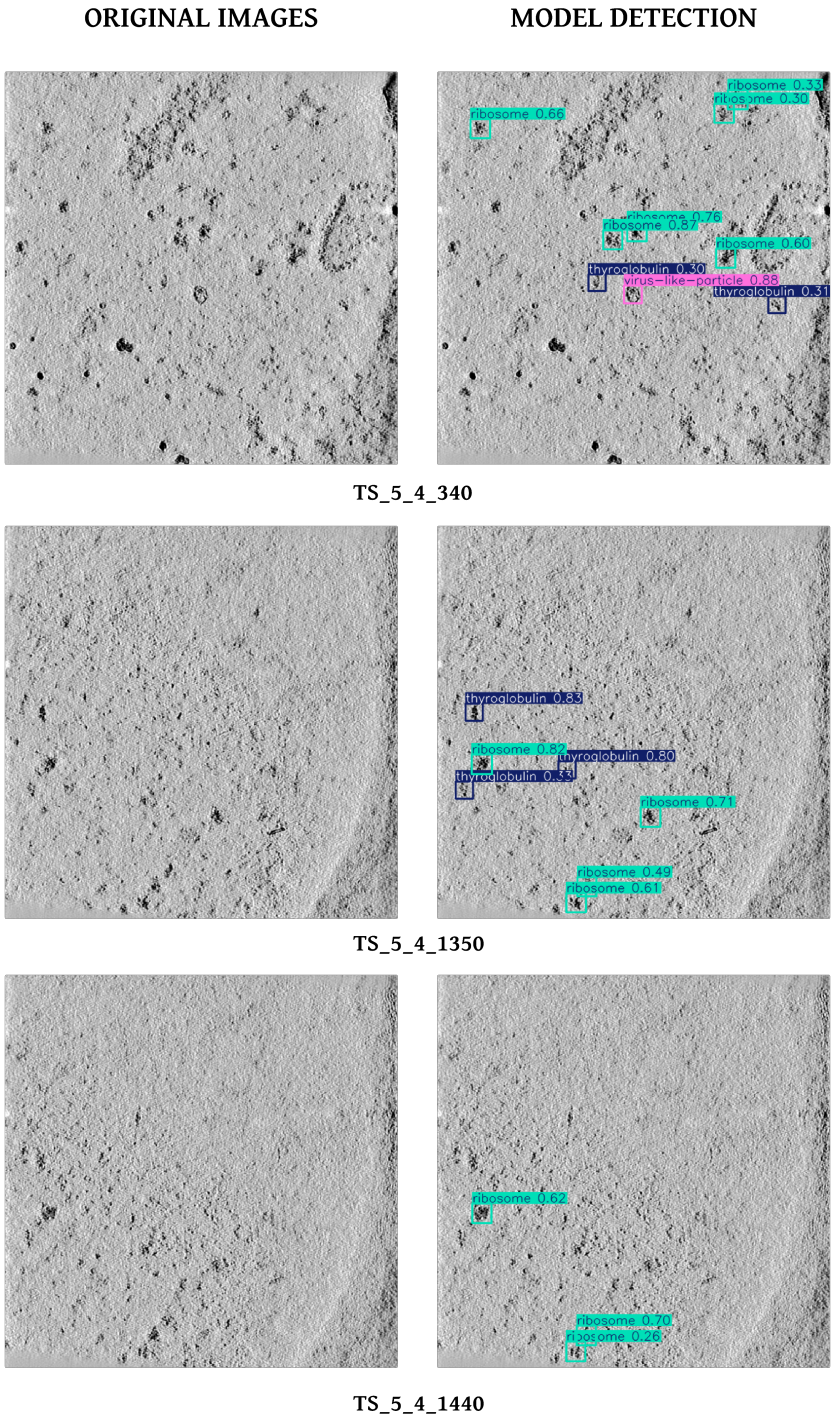

Figure 7. Validation Dataset Prediction with YOLO

D. 3D U-NET Model Architecture

The 3D U-Net architecture, depicted in Figure 8, is a volumetric segmentation model designed for analyzing 3D data, such as tomograms. It leverages an encoder-decoder structure with skip connections to preserve spatial information during the downsampling and upsampling processes. This architecture is particularly effective for handling sparse 3D data and capturing context across multiple scales.

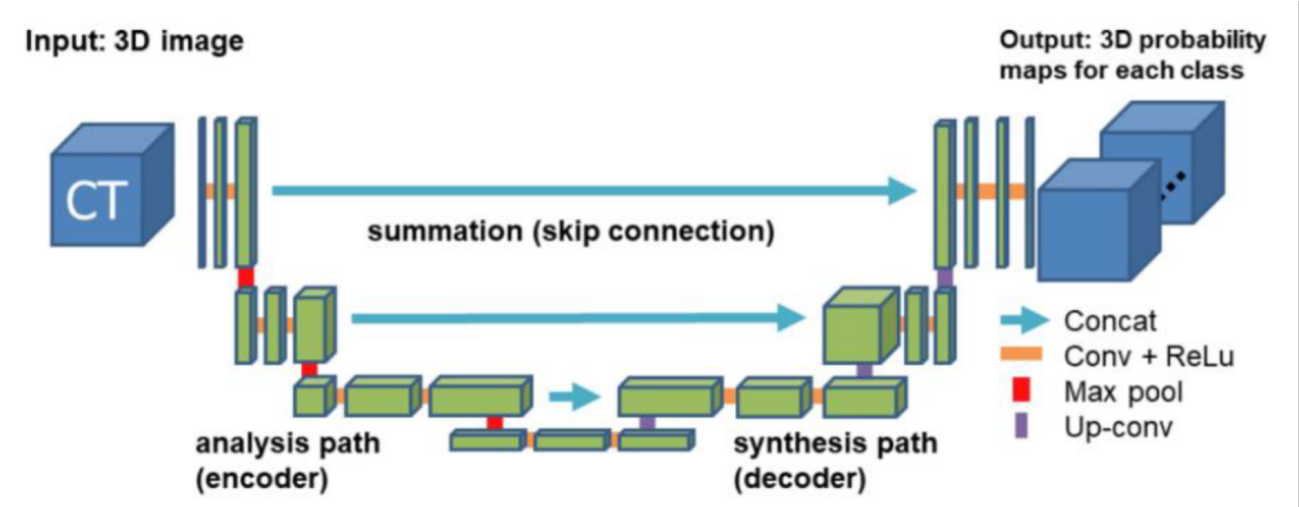

Figure 8. 3D U-Net Architecture

E. 3D U-NET Evaluation of test dataset

Table 3 provides detailed performance metrics for each particle type identified by the model during evaluation on data TS_6_4. The metrics are used to assess the precision, recall, and F-beta score, which collectively evaluate the effectiveness of particle identification, with an emphasis on recall due to the higher weight assigned to missed particles in the F-beta formula.
Columns description:

1. **Particle Type**: The type of particle being evaluated, including apo-ferritin, beta-amylase, beta-galactosidase, ribosome, thyroglobulin, and virus-like-particle.

2. **P** (Predicted): The number of particles predicted by the model for the given particle type.

3. **T** (True): The actual number of ground-truth particles for the given type.

4. **Hit**: The number of particles correctly identified by the model within the acceptable radius.

5. **Miss**: The number of true particles missed by the model.

6. **FP** (False Positives): The number of particles incorrectly predicted by the model.

7. **Precision**: The proportion of predicted particles that are correct.

8. **Recall**: The proportion of true particles that are correctly identified.

9. **F-beta=4**: The F-beta score with a beta value of 4, emphasizing recall over precision.

10. **Weight**: The importance assigned to each particle type, where harder particles (e.g., beta-galactosidase and thyroglobulin) receive higher weights.

Particle Type	P	T	Hit	Miss	FP	Precision	Recall	F-beta=4	Weight
apo-ferritin	66	139	51	88	15	0.772727	0.366906	0.378603	1
beta-amylase	98	31	18	13	80	0.183673	0.580645	0.515152	0
beta-galactosidase	157	40	22	18	135	0.140127	0.550000	0.469260	2
ribosome	290	142	106	36	184	0.365517	0.746479	0.703357	1
thyroglobulin	560	94	83	11	477	0.148214	0.882979	0.683624	2
virus-like-particle	96	30	28	2	68	0.291667	0.933333	0.826389	1

Table 3. Performance metrics for each particle type on data TS_6_4

And the **Final lb_score** of the model is 0.6020165387797923.

The following visualizations Figure 9 illustrate the model's performance in detecting particles on real-world data TS_6_4. These visualizations help assess the alignment of the model's predictions with the ground truth and highlight common failure cases.

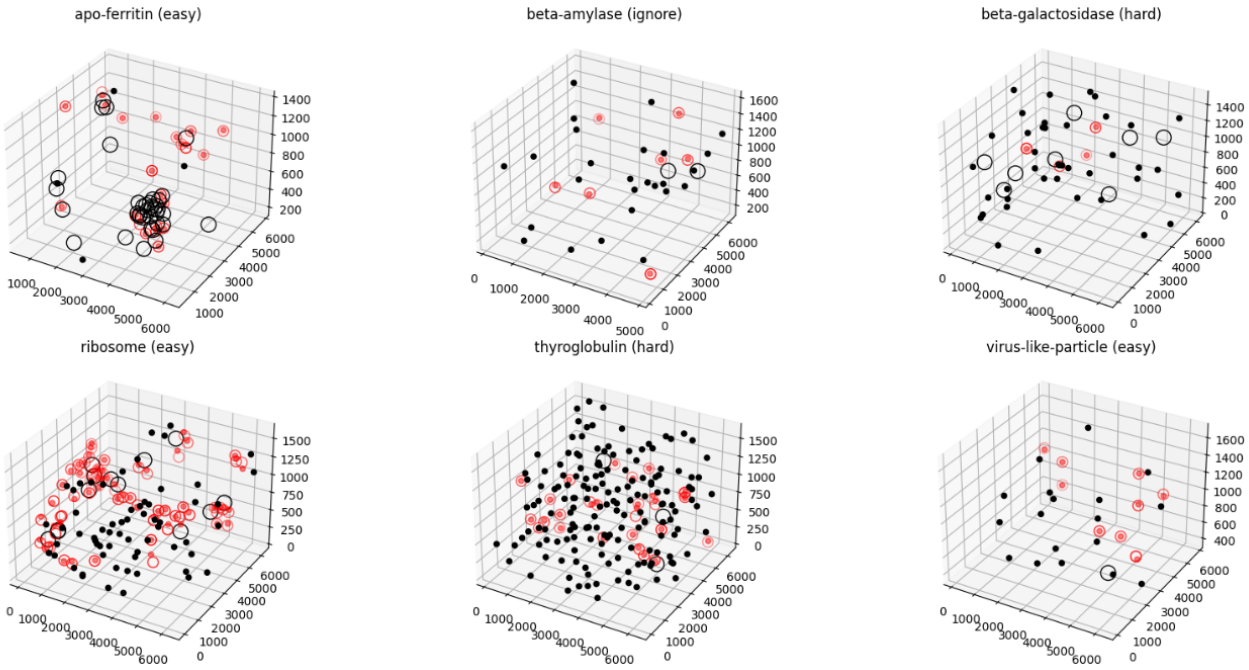

Figure 9. Visualization of the Detection of the model on data TS_6_4

F. YOLO11 Pre-Trained Model Architecture

The following image shows the pre-trained architecture of YOLO11. It is composed of three components bases on Backbone, Neck and Head.

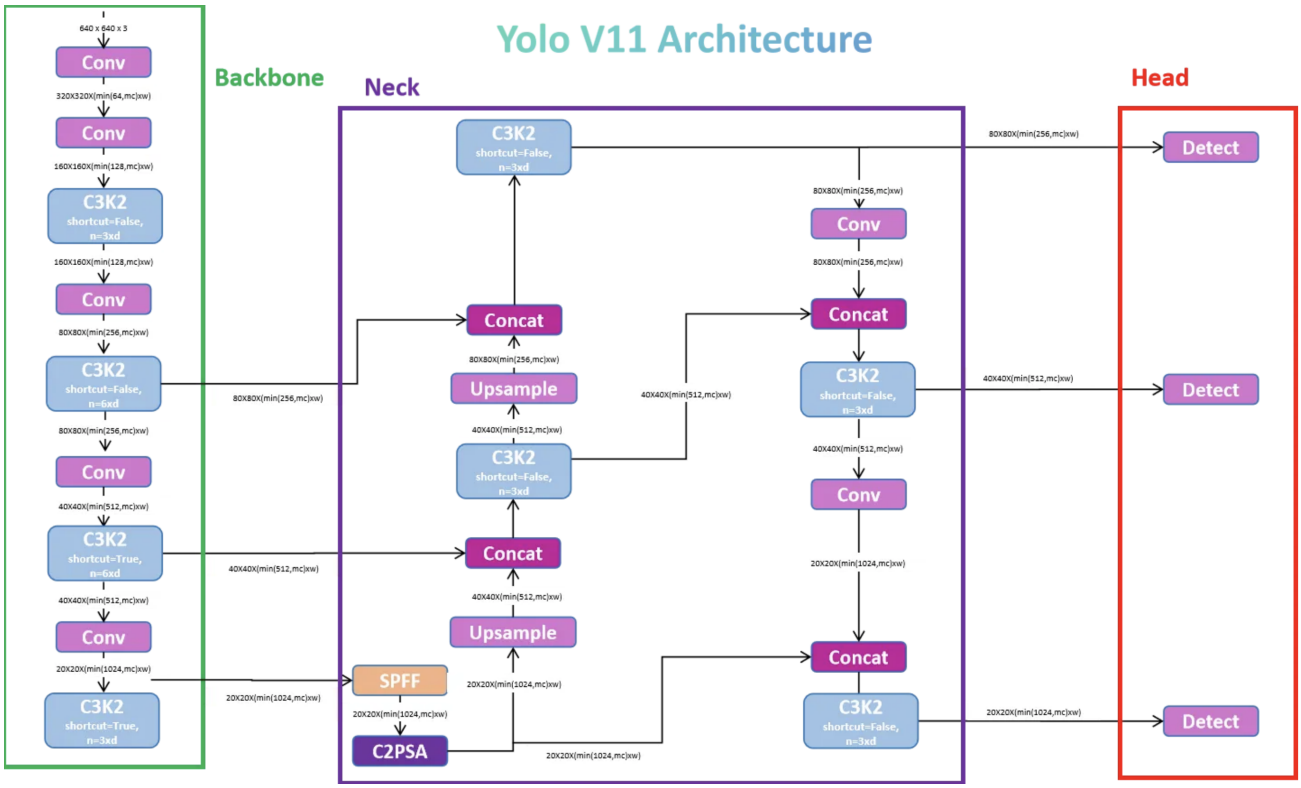

Figure 10. YOLO11 Architecture

G. YOLO11 Prediction Matrix

The following image represents a normalized confusion matrix from the results of a model training, each component in the matrix is a labeled particle type in the model and also includes a type ”Background” which is not a particle type but the backbone of the image.

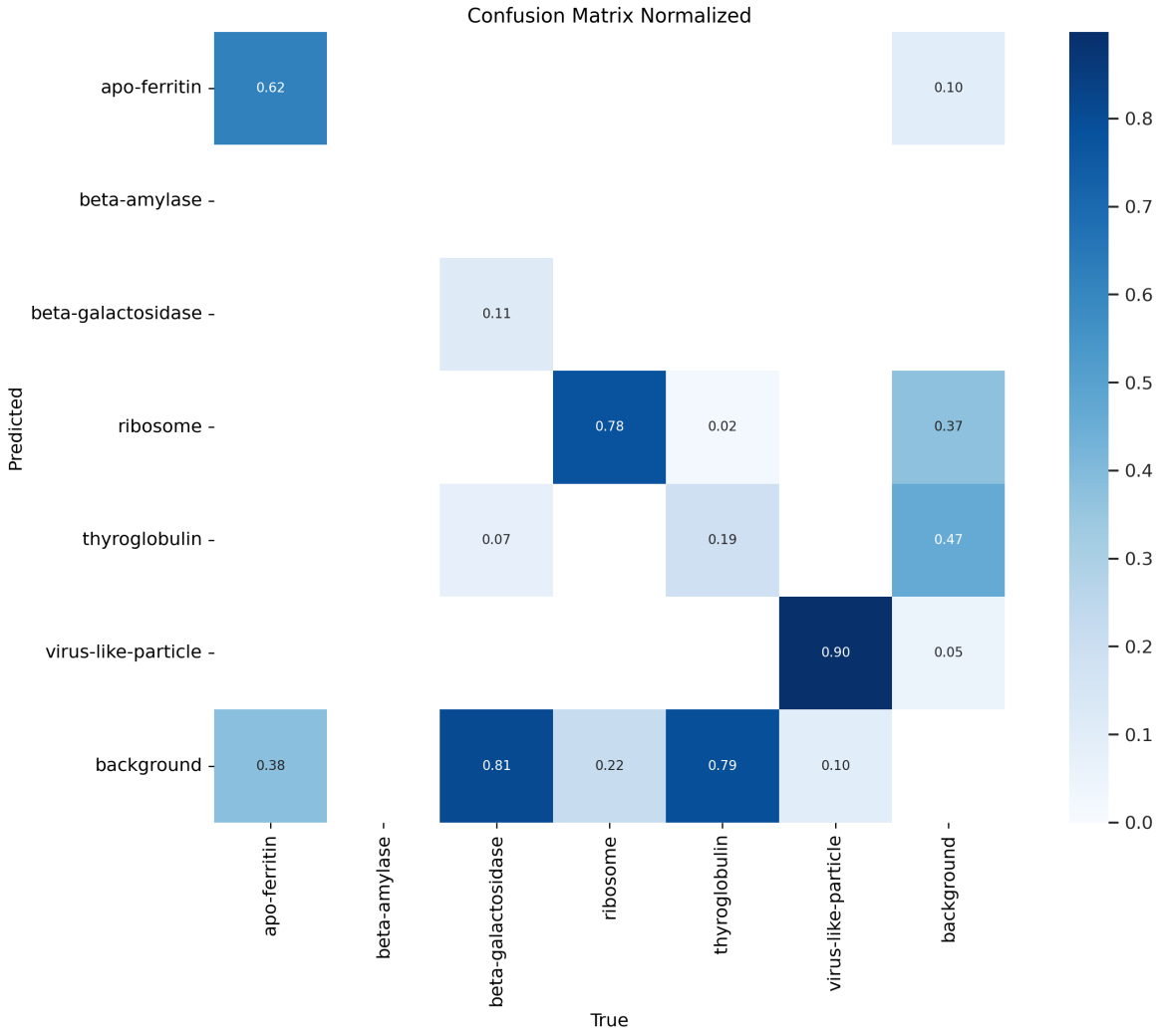

Figure 11. Normalized Confusion Matrix from YOLO11 Training