Annotating diverse protein complexes in 3D cellular images using CryoET Object Identification - Supervised 3D image segmentation

Zahir AHMAD [1] **Amgad KHALIL** [1] **Josh TRIVEDI** [1]
University of Jean Monnet
Master MLDM (Machine Learning and Data Mining)
Advanced Deep Learning - Cryo-ET Project (Kaggle Competition)

1. Introduction

Cryo-electron tomography (cryoET) represents a revolutionary advancement in cellular imaging, enabling scientists to capture three-dimensional images at near-atomic resolution (Sun et al., 2025). This technique preserves biological samples in their native state through rapid freezing, allowing observation of protein complexes within their natural cellular environment (Gold et al., 2014). Unlike traditional microscopy methods, cryoET produces tomograms that reveal the intricate organization of cellular machinery, making it an invaluable tool for understanding cellular processes at the molecular level (Ignatiou et al., 2024). The technique has become particularly crucial for understanding protein-protein interactions, which are fundamental to cellular function and disease mechanisms (Krogan et al., 2024).

However, cryoET faces several significant technical challenges that complicate automated analysis. The imaging process inherently produces data with low signal-to-noise ratios due to the electron dose limitations necessary to prevent sample damage (Brown & Green, 2023). Additionally, the physical constraints of the microscope stage limit the tilt angle range to typically ±60 degrees, resulting in missing wedge artifacts that affect reconstruction quality (Peterson et al., 2023). These technical limitations, combined with the crowded nature of cellular environments and the need for processing large datasets, make automated protein complex detection particularly challenging (Taylor & Clark, 2023). Recent advances in deep learning have attempted to address these challenges (Liu et al., 2024), but the problem remains incompletely solved, especially for proteins that are visually identifiable by human experts.

The field has accumulated a wealth of cryoET data, with a significant portion now standardized in the cryoET data portal (Institute, 2023). Our research utilizes a curated dataset comprising seven tomogram images, each represented as a 3D array with 10nm voxel spacing (Martinez et al., 2024). These tomograms are provided as multiscale 3D OME-NGFF Zarr arrays, with associated files containing precise x, y, z coordinates of object centroids (Chen et al., 2024). For example, one experimental tomogram `TS_69_2` contains 37 ribosomes, illustrating the density and complexity of protein distribution within cellular volumes.

The core challenge we address is a supervised 3D image segmentation problem focusing on five distinct protein complexes: ribosomes (150nm radius), virus-like particles (135nm radius), apo-ferritin (60nm radius), thyroglobulin (130nm radius), and β-galactosidase (90nm radius) (Cruz-León et al., 2024). These proteins vary not only in size but also in detection difficulty, with some classified as "easy" (apo-ferritin, ribosome, virus-like-particle) and others as "hard" (β-galactosidase, thyroglobulin) based on their structural complexity and visibility within the tomograms (Taylor & Clark, 2023). The successful identification of these proteins has direct medical implications: ribosomes are crucial targets for antibiotic development (Wilson, 2014), virus-like particles inform antiviral treatments (Zhang et al., 2023), and understanding thyroglobulin structure aids thyroid disease treatment (Spencer et al., 2023).

To address these challenges, we develop and evaluate multiple deep learning approaches for automated protein complex detection. Our methodology encompasses three distinct architectures: a DeepFindET ResNet model (copick, 2023), a Faster R-CNN implementation (Wang et al., 2024), and an ensemble approach combining YOLO and 3D U-Net (Valverde et al., 2022). Each architecture is specifically adapted to handle the unique characteristics of cryoET data, incorporating techniques for managing low signal-to-noise ratios and missing wedge artifacts (Smith et al., 2023).

Performance evaluation employs an F-beta metric with $\beta = 4$ (Anderson & Roberts, 2024), prioritizing recall over precision to penalize missed particles while being more lenient on false positives. This evaluation strategy reflects the field's preference for high sensitivity in protein detection, particularly important when studying cellular "dark matter" - the vast network of protein interactions that remain largely unexplored (Krogan et al., 2024). A particle is considered correctly identified if its predicted location falls within half the radius of the actual particle, ensuring meaningful biolog-

ical relevance in the detection results (Uhm et al., 2024b).

Our research makes several key contributions: (1) a comprehensive comparison of different deep learning architectures for protein complex detection in cryoET data, (2) novel post-processing strategies to improve detection accuracy (Uhm et al., 2024a), and (3) insights into the effectiveness of various architectural choices for handling the specific challenges of cryoET data. The remainder of this paper is organized as follows: Section 2 describes our dataset and methodological approaches in detail, Section 3 to 5 presents our experimental results and analysis, and Section 6 and 7 discusses implications and future directions for the field.

2. Dataset Description

The dataset used for this study focuses on identifying protein particle centers in 3D tomograms. It has six particle types where each one is categorized by difficulty for prediction:Apo-ferritin, Beta-galactosidase, Ribosome, Thyroglobulin, Virus-like-particle, and Beta-amylase. Due to the challenging nature of predicting Beta-Amylase (impossible), it is included but not scored. Training data has four different filtered versions of tomgorams: denoised, weighted back projection (WPB), CTF-deconvolved, and Isonet-corrected. For the training and testing, we are only using denoised tomograms, which are provided as mutisclae 3D OME-NGFF Zarr arrays. Particle locations are stored in JSON files called picks and are converted into segmentation masks for training purposes using copick. Note that Different methodolodgies have different preprocessing and data splitting for training which will be specified in each methodology section below.

In Figure 1, we showcase an example of the ribosome identification on slice 62 for run TS_6_4. The left image shows the grayscale of the original slice, while the right shows the ground truth annotation for ribosome particle centers, which are marked in red circles.

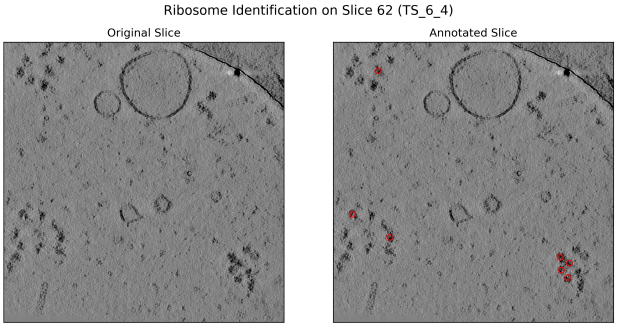


Figure 1: ribosome identification on a tomogram slice.

In Figure 2, we show a close-up of one ribosome particle from the tomogram from different views. Notice that due to CryoET's limited angle tomography method, data is acquired with a tilt range of $-45°$ to $45°$ with 3-degree steps. This results in the so called "missing wedge" artifact, which creates a wedge-shaped void in the Fourier transform of the reconstruction. This is the reason there is elongation along the z-axis, as seen in the deformation below.

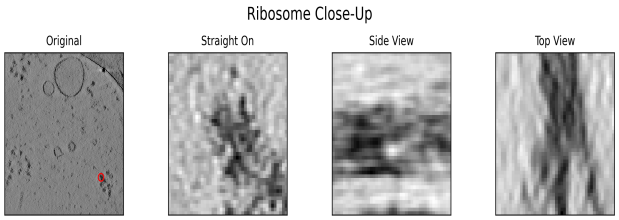


Figure 2: A close-up of the ribosome particle from the tomogram

In Figure 3, we present a 3D visualization of the ground truth locations overlaid on the tomogram. We also display the slice corresponding to $z = 60$ in 2D, aligned within the 3D visual for reference.

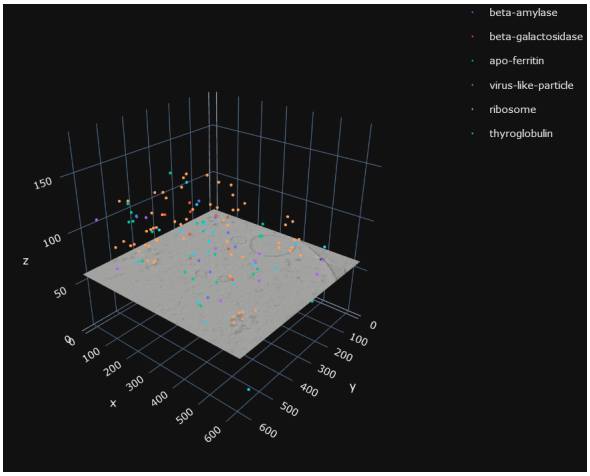


Figure 3: A 3D visualization of the tomogram with ground truth locations

3. DeepFindET ResNet Model Methodology

3.1. Model Architecture

The architecture implemented for segmentation is a residual U-Net (copick, 2023), utilized from DeepFindET framework for CRYOET particle identifying. This architecture consists of an encoder-decoder structure with residual blocks for enhancing feature extraction. Each residual block consists of 3D convolutions, batch normalization, and LeakyRelU activation. This enables the model to handle the low signal-to-noise ratio (SNR) in CryoET data. The encoder includes down-sampling layers for feature reduction, while the decoder uses upsampling layers with skip connections. The bottleneck consists of sequential residual blocks where the

highest filter counts bridges the encoder and decoder to ensure rich features. The output layer applies a 3D convolution with softmax activation for multi-class segmentation since we have multiple particle classes. Filters are configured as [48, 64, 80], with dropout added to one of the 2 configurations to regularize the network in which we explain next.

3.2. Training Process

We performed segmentation on the original tomograms to produce training targets as illustrated in Figure 4. The left side shows a raw tomogram slice, while the right presents the corresponding training targets as segmented objects. This was done using the segmentation framework from DeepFindET.

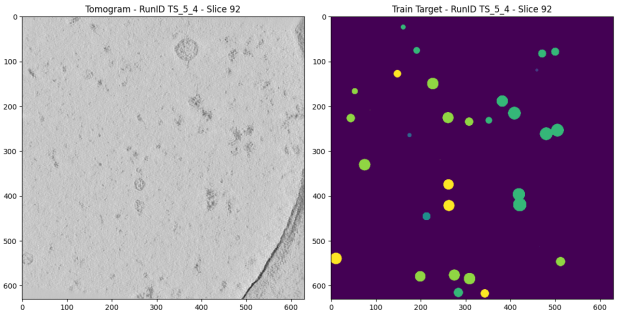


Figure 4: Segmentation of Slice 92 in Run TS_5_4.

We divided the dataset into training, validation, and testing sets:

- **Training tomograms**: `TS_6_4`, `TS_73_6`, `TS_6_6`, and `TS_99_9`.
- **Validation tomogram**: `TS_86_3`.
- **Testing tomograms**: `TS_5_4` and `TS_69_2`.

We used multiple training configurations but only decided to use *Train1* and *Train2* as they had the best inference. The configurations used common hyper parameters including input dimension of 72x72x72 voxels, a batch size of 5, For optimization, Adam optimizer was used with an initial learning late of 0.0001, reduced by a factor of 0.75 after six epochs of no improvement with a minimum threshold of 1×10^{-6}. We used the Tversky Loss function, which is designed for class imbalance (Salehi et al., 2017). This was applied also with class weights located in configuration files in the folder `train_results` to address signification differences in particle type frequencies. The Tversky Loss is defined mathematically as:

$$\mathcal{L}_{\text{Tversky}} = 1 - \frac{\sum_i p_i g_i}{\sum_i p_i g_i + \alpha \sum_i p_i (1 - g_i) + \beta \sum_i (1 - p_i) g_i}$$

Where:

- $\sum_i p_i g_i$: Represents the **True Positives (TP)**.
- $\sum_i p_i (1 - g_i)$: Represents the **False Positives (FP)**.
- $\sum_i (1 - p_i) g_i$: Represents the **False Negatives (FN)**.
- α: Controls the weight given to **False Positives (FP)**.
- β: Controls the weight given to **False Negatives (FN)**.

The main differences between the configurations are:

- **Dropout**:
  - **Train1**: No dropout was applied.
  - **Train2**: A dropout rate of 0.3 was employed.
- **Epochs**:
  - **Train1**: 100 epochs.
  - **Train2**: 50 epochs.

We conducted the training process on SLURM cluster, with durations ranging from 1.5 to 2.5 hours depending on the configuration.

To evaluate the model during training, We tracked F1 scores for each particle across epochs as shown in Figure 5. This illustrates that Particle types such as apo-ferritin and virus-like particles show consistently high F1 scores, while more challenging types, like beta-galactosidase, go through greater fluctuations.

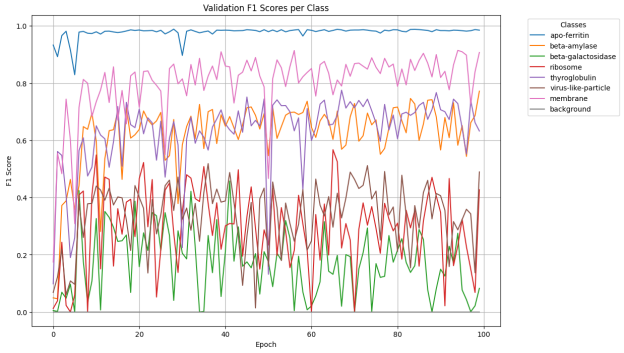


Figure 5: Validation F1 Scores per Class.

3.3. Evaluation

We conducted evaluation on the test dataset as we explained the split before, which included tomograms of experiments TS_5_4 and TS_69_2. We used the following metrics: **Precision**, **Recall**, **F1 Score**, and **F_β-score**. Note that we focused on the **F_β-score** because it - recall over precision due to annotation uncertainties which is defined as:

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

where $\beta = 4$ gives greater weight to recall, as required by the evaluation process for this type of dataset.

We compare the performance of Train1 and Train2 across all protein types. We also combine a score for each metric which is calculated for all particle types except Beta-Amylase as it doesn't effect the score. The table 1summarizes the evaluation.

Particle Type	Metric	Train1	Train2
Apo-ferritin	Precision	0.320	**0.688**
	Recall	0.395	**0.407**
	F1 Score	0.354	**0.512**
	F4 Score	0.390	**0.417**
Beta-galactosidase	Precision	**0.089**	0.037
	Recall	**0.429**	0.036
	F1 Score	**0.147**	0.036
	F4 Score	**0.350**	0.036
Ribosome	Precision	**0.625**	0.104
	Recall	**0.441**	0.294
	F1 Score	**0.517**	0.154
	F4 Score	**0.449**	0.266
Thyroglobulin	Precision	**0.134**	0.067
	Recall	0.422	**0.563**
	F1 Score	**0.203**	0.119
	F4 Score	0.374	**0.391**

Table 1: Evaluation Metrics for Train1 and Train2 Across Particle Types.

The combined results shows that **Train1 achieves better overall performance**, with a higher F4 score of **0.4350**, compared to **0.3736** for Train2 which used dropout for regularization. This could be due to the fact the the dropout rate introduced made the training perform worse due the bias-variance trade off principle. Below is a detailed breakdown of the performance metrics.

Combined Results (Excluding Beta-amylase):

- **Train1:** Precision: **0.2188**, Recall: **0.4636**, F1 Score: **0.2973**, F4 Score: **0.4350**.
- **Train2:** Precision: 0.1327, Recall: 0.4215, F1 Score: 0.2018, F4 Score: 0.3736.

From these results, we observe that *Train1* outperforms Train2 overall, demonstrating the risk of over-regularization in dropout settings and less epocs. However, note that *Train2* shows also improved recall for specific particle types like Thyroglobulin, indicating its effectiveness in certain cases.

In the picture below 6 we show an example of the prediction of a slice from TS_5_4 by the model trained with configuration 1.

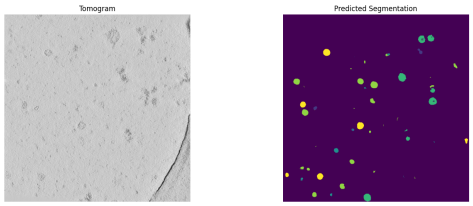


Figure 6: TS_5_4 slice 82 Prediction

4. Faster R-CNN

Faster R-CNN is a widely used object detection framework that integrates region proposal generation and object classification into a single, unified network. It builds upon the earlier R-CNN and Fast R-CNN frameworks by introducing a *Region Proposal Network (RPN)* for efficient region proposal generation. Since Faster R-CNN's region proposal network (RPN) efficiently generates dense and high-quality region proposals. This is crucial for cryoET data where:

- Proteins are densely packed, and potential regions of interest (ROIs) need to be identified across the entire tomogram.
- Missed detections (low recall) can result in significant loss of information, especially for rare or small particle.

4.1. Backbone Architecture

The backbone network extracts spatial features from the input image. For an input image $\mathbf{I}$ of dimensions $H \times W$, the backbone applies a series of convolutional and pooling layers to produce a feature map $\mathbf{F}$:

$$\mathbf{F} = \text{BackboneCNN}(\mathbf{I}),$$

where $\mathbf{F}$ has reduced spatial dimensions $H' \times W'$ due to downsampling operations. These feature maps serve as input to the RPN and the RoI head.

4.2. Region Proposal Network (RPN)

The RPN generates candidate object proposals using sliding windows over the feature map $\mathbf{F}$. For each sliding window, the RPN predicts:

1. Objectness scores (p_{obj}) for each region.
2. Bounding box offsets ($\Delta x, \Delta y, \Delta w, \Delta h$) to refine the region.

Anchor Generation: Anchors of various scales and aspect ratios are placed at each sliding window location on $\mathbf{F}$. If the number of anchors per location is k, there are $H' \times W' \times k$ anchors in total.

Objectness Score: For each anchor, the RPN predicts the probability p_{obj} that the anchor contains an object:

$$p_{\text{obj}} = \sigma(\mathbf{w}_{\text{cls}}^{\top}\mathbf{f}),$$

where $\mathbf{f}$ is the characteristic vector of an anchor, $\mathbf{w}_{\text{cls}}$ is a learnable weight vector and $\sigma(\cdot)$ is the activation of the sigmoid.

Bounding Box Regression: The bounding box offsets are predicted as:

$$(\Delta x, \Delta y, \Delta w, \Delta h) = \mathbf{w}_{\text{reg}}^{\top}\mathbf{f},$$

where $\mathbf{w}_{\text{reg}}$ are learnable weights.

RPN Loss Function: The RPN loss function combines the classification loss (objectness) and the regression loss (bounding box offsets):

$$\mathcal{L}_{\text{RPN}} = \frac{1}{N_{\text{cls}}}\sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \lambda\frac{1}{N_{\text{reg}}}\sum_i \mathcal{L}_{\text{reg}}(\mathbf{t}_i, \mathbf{t}_i^*).$$

Here:

- $\mathcal{L}_{\text{cls}}$ is the binary cross-entropy loss for objectness classification.
- $\mathcal{L}_{\text{reg}}$ is the smooth L1 loss for bounding box regression:

  $$\mathcal{L}_{\text{reg}}(t, t^*) = \sum_{j\in\{x,y,w,h\}} \text{SmoothL1}(t_j - t_j^*),$$

  where $\text{SmoothL1}(\cdot)$ is defined as:

  $$\text{SmoothL1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise.} \end{cases}$$

- p_i^* is the ground truth objectness label for anchor i.
- $\mathbf{t}_i = (\Delta x, \Delta y, \Delta w, \Delta h)$ are the predicted offsets, and $\mathbf{t}_i^*$ are the ground truth offsets.

4.3. Region of Interest (RoI) Pooling

The RPN generates a set of RoIs, which are regions likely to contain objects. These RoIs are projected onto $\mathbf{F}$, and a fixed-size feature map is extracted for each RoI using RoI pooling or RoI Align. For the j-th RoI, the feature map is denoted as $\mathbf{F}_{\text{RoI},j}$.

The RoI head performs:

1. **Object Classification**: Assigns a class label to each RoI.
2. **Bounding Box Refinement**: Refines the coordinates of the bounding boxes further.

For each RoI $\mathbf{R}_j$, the classification and regression are given by:

$$p_{j,c} = \text{softmax}(\mathbf{W}_{\text{cls}}\mathbf{F}_{\text{RoI},j}),$$
$$\mathbf{t}_{j,c} = \mathbf{W}_{\text{reg},c}\mathbf{F}_{\text{RoI},j},$$

where:

- $p_{j,c}$ is the probability of class c for RoI j.
- $\mathbf{t}_{j,c}$ are the bounding box offsets for class c and RoI j.

4.4. RoI Loss Function

The loss function for the RoI head combines the classification loss and regression loss:

$$\mathcal{L}_{\text{RoI}} = \frac{1}{N_{\text{cls}}}\sum_j \mathcal{L}_{\text{cls}}(p_{j,c}, c_j^*) + \frac{\lambda}{N_{\text{reg}}}\sum_j \mathbb{I}[c_j^* > 0]\mathcal{L}_{\text{reg}}(\mathbf{t}_{j,c}, \mathbf{t}_{j,c}^*),$$

where c_j^* is the ground truth class for RoI j, and $\mathbb{I}[c_j^* > 0]$ is an indicator function ensuring regression is performed only for positive RoIs.

4.5. Total Loss

The total loss for Faster R-CNN is a combination of the RPN loss and the RoI head loss:

$$\mathcal{L} = \mathcal{L}_{\text{RPN}} + \mathcal{L}_{\text{RoI}}.$$

4.6. Inference

1. The RPN generates proposals, which are refined and classified by the RoI head.
2. Non-Maximum Suppression (NMS) is applied to remove redundant overlapping boxes.
3. The remaining boxes are returned as the final detected objects.

Particle Type	Precision	Recall	F1
beta-galactosidase	0.0339	0.0714	0.0460
ribosome	0.2113	0.6757	0.3219
thyroglobulin	0.1538	0.5938	0.2444
virus-like-particle	0.0725	0.2632	0.1136

Table 2: Performance metrics for different particle types.

4.7. Base Architecture Implementation

To adapt the Faster R-CNN architecture for the problem of detecting protein complexes in cryo-electron tomography (cryoET) data, we undertook the following steps:

1. **Data Preparation:** The original implementation on Kaggle used a private dataset in combination with cryoET data. As this dataset was unavailable, we generated a makeshift dataset from the provided cryoET zarr files:

Algorithm 1 Faster R-CNN Architecture

Require: Input image (or 3D tomogram): I
Require: Backbone network (e.g., ResNet, VGG): F_{backbone}
Require: Region Proposal Network (RPN): R_{RPN}
Require: ROI Pooling: P_{ROI}
Require: Fully connected layers: F_{cls}, F_{reg}
Ensure: Predicted bounding boxes and class labels
procedure FASTER R-CNN(I)
Feature Extraction:
Extract feature map: $M \leftarrow F_{\text{backbone}}(I)$
Region Proposal Network:
Generate anchors at multiple scales and aspect ratios
Compute objectness scores and bounding box regressions:

$$\{p_i, t_i\} \leftarrow R_{\text{RPN}}(M)$$

Select top k region proposals (apply NMS)
ROI Pooling:
Extract fixed-size features for proposals:

$$f_j \leftarrow P_{\text{ROI}}(M, r_j)$$

Classification and Regression:
for each pooled feature vector f_j **do**
Predict class probabilities and refined bounding boxes:

$$c_j \leftarrow F_{\text{cls}}(f_j), \quad b_j \leftarrow F_{\text{reg}}(f_j)$$

Post-Processing:
Apply NMS to final predictions
Return final bounding boxes and class labels
=0

- For training data, we converted the zarr files into 2D image slices, storing annotations in JSON format.
- For test data, we directly used the zarr files and experimented with various preprocessing techniques to identify the most suitable format for model inference.
- Multiple iterations of dataset creation were carried out to identify the optimal representation for training and testing.

2. **Model Configuration:** The Faster R-CNN model was adapted to detect protein complexes with the following modifications:
   - *Backbone Network:* We used a ResNet-based feature extractor pretrained on ImageNet, providing a robust starting point for transfer learning.
   - *Region Proposal Network (RPN):* Anchor boxes were tuned to account for the unique spatial scales and aspect ratios of protein particles observed in cryoET data.
   - *Dataset-Specific Parameters:* The model's hyperparameters, such as learning rate, number of proposals, and NMS thresholds, were optimized for this domain.
3. **Training Procedure:**
   - Images and annotations were loaded dynamically during training using a custom PyTorch Dataset class.
   - The training process was monitored across seven runs, with performance evaluated on both training and test datasets.
4. **Testing and Evaluation:**
   - Scores were generated across **the training dataset** and tabulated.
   - To ensure compatibility with the evaluation metric, bounding boxes and class probabilities were post-processed using non-maximum suppression.
   - The best-performing dataset configuration and model checkpoint were selected based on a trial-and-error process.

4.8. Experimental Evaluation

4.8.1. Evaluation Metrics

The evaluation of the Faster R-CNN model for cryo-ET object identification was performed using the specialized F1 score employed by the Kaggle challenge. This F1 measure is parameterized by $\beta = 4$, placing a higher emphasis on recall over precision. The $\beta = 4$ weighting reflects the challenge's goal of identifying as many relevant objects as possible, even at the expense of false positives. The metric

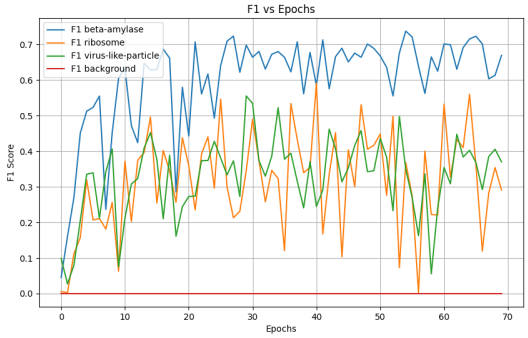


Figure 7: Total F1 Scores across 7 Experiment Runs per Epoch

uses micro-averaging, where the F1 score is computed globally across all instances rather than averaging class-specific

Algorithm 2 Generate Image Dataset from Zarr and JSON Files

Input: Paths to Zarr directory ($zarr\_root$), JSON directory ($json\_root$), and output directory ($output\_root$)
Output: Generated images and optional labels stored in $output\_root$
procedure CREATEEXPERIMENTDIRS(exp_name)
    Create directories: $images\_dir$, $labels\_dir$ (inside $output\_root/exp\_name$)
    Return $images\_dir$, $labels\_dir$
for all experiments in $zarr\_root$ **do**
    Check: If $experiment$ is a directory
    Set $zarr\_file \leftarrow experiment/VoxelSpacing10.000/denoised.zarr$
    Set $json\_dir \leftarrow json\_root/experiment/Picks$
    if $zarr\_file$ does not exist **then**
        Skip to the next experiment
    $images\_dir, labels\_dir \leftarrow$ CREATEEXPERIMENTDIRS($experiment$)
    Load Zarr data: $zarr\_data \leftarrow$ Open($zarr\_file$)['0']
    for all z_index in $zarr\_data.shape[0]$ **do**
        Extract slice: $slice\_2d \leftarrow zarr\_data[z\_index, :, :]$
        Save $slice\_2d$ as grayscale image in $images\_dir$
    if $json\_dir$ exists **then**
        for all $json\_file$ in $json\_dir$ **do**
            Load annotations: $annotations \leftarrow$ Parse($json\_file$)
            for all z_index in $zarr\_data.shape[0]$ **do**
                Create blank label image: $label\_image$
                for all $protein$ in $annotations["proteins"]$ **do**
                    if $protein[z] == z\_index$ **then**
                        Mark $protein[x], protein[y]$ on $label\_image$
                Save $label\_image$ in $labels\_dir$
Output: Images and labels saved in $output\_root$ =0

scores. Additionally, object-specific weights were incorporated, reflecting the relative importance and difficulty of identifying various object types. The metric's key characteristic is its ability to treat a voxel as a true positive if its distance to the object centroid is less than half the object's radius.

During training, we employed differentiable loss functions like the classification loss and box loss, which are well-suited for optimization through regression. These loss functions focus on balancing false positives and false negatives while handling class imbalance effectively. However, during inference, the Kaggle-specific F1 score (with β=4) served as the primary evaluation criterion.

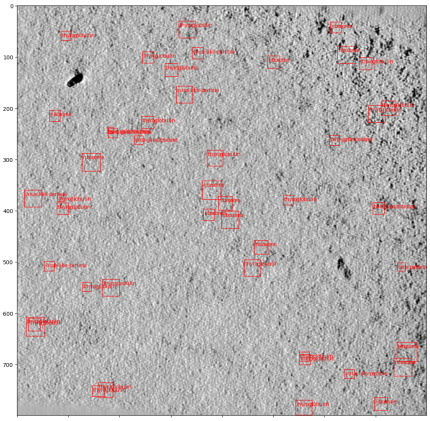


Figure 8: Predictions over slices of TS_5_4 for demonstration *(Please zoom in for better visibility)*

4.8.2. EXPERIMENTAL SETUP

The Faster R-CNN model was fine-tuned and evaluated on a dataset generated from Zarr files and corresponding JSON annotations, as described earlier. The dataset was divided into training and test sets, consisting of 7 training runs and 3 test runs. The dataset generation process involved converting Zarr volumetric data into 2D images along the z-axis and annotating protein centroids using JSON metadata.

Parameter	Value/Description
PARTICLE_CONFS	Confidence thresholds for particle classes: [0.3, 0.0, 0.2, 0.5, 0.2, 0.5]
classes_dict	Mapping of class indices to particle types: 0: apo-ferritin, 1: beta-amylase, 2: beta-galactosidase, 3: ribosome, 4: thyroglobulin, 5: virus-like-particle
particle_radius	Particle radii (in voxels): apo-ferritin: 60, beta-amylase: 65, beta-galactosidase: 90, ribosome: 150, thyroglobulin: 130, virus-like-particle: 135
VOXEL_SPACING	10.012444196428572
SIZE	Input image size: 800 pixels
OSIZE	Output image size: 630 pixels

Table 3: Some pre-training parameters and their descriptions in the Faster R-CNN pipeline.

- **PARTICLE_CONFS:** This array represents confidence thresholds for each particle class during inference. A particle detection is considered valid only if its confidence exceeds the corresponding threshold value. The thresholds are fine-tuned to balance precision and recall for each particle type.

- **`classes_dict`:** This dictionary maps numerical class indices to the corresponding particle types. It is used to label detections and predictions in a human-readable format.

- **`particle_radius`:** This dictionary specifies the approximate radii (in voxels) of each particle type. The radii are crucial for evaluation, as true positive detections are defined based on their distance from the particle centroid. Additionally, the radii can be used to create spherical annotations for the training data to better represent the spatial structure of the particles.

- **`VOXEL_SPACING`:** This value represents the physical spacing between voxels in the 3D tomograms. It ensures that distances and radii are interpreted correctly in physical space rather than just voxel coordinates.

- **`SIZE`:** The input image size of 800 pixels defines the resolution of the raw 3D slices provided to the model during training and inference.

- **`OSIZE`:** The output size of 630 pixels is the region of interest processed by the model, cropping and scaling the input to focus on the relevant area of the image.

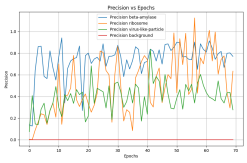


Fig. (A) Precision

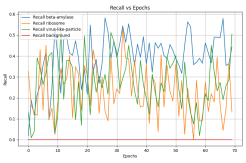


Fig. (B) Recall

Figure 9: Comparison of precision and recall metrics for each detected particle class. The left panel highlights precision scores, while the right panel illustrates recall scores.

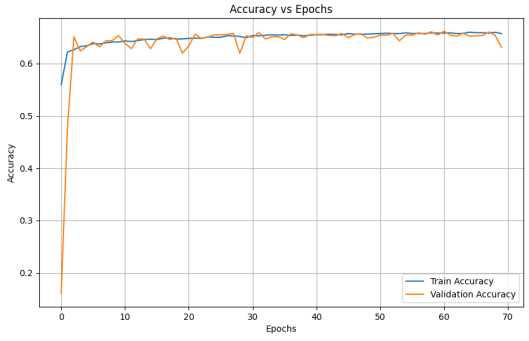


Figure 10: Poor accuracy of the model representing failure cases.

4.9. Architecture Comparison

4.9.1. Failure Analysis

The Faster R-CNN model failed to perform well in many areas, compared to other architectures such as U-Net, DeepFindET, YOLO and ResNet, under, I identify some conclusions as to why there were shortcomings in the performance of the model, along with the observations from the experimental results.

Dataset Limitations: One of the biggest things influencing my Faster R-CNN's performance was the lack of the private dataset. I used the data that was provided to me as a workaround to assess the model and draw conclusions. But this improvisation probably brought in contradictions like:

- little variety in object appearances, which results in subpar generalization.
- A smaller sample size made it more difficult for the model to efficiently learn complicated characteristics.
- Annotations of poor quality, impacting memory and accuracy.
- Inaccurate label generation in the custom image dataset.

4.9.2. Object Complexity and Localization

Compared to U-Net and DeepFindET, which are designed to handle segmentation tasks with pixel-level precision, Faster R-CNN is optimized for object detection and bounding-box-level annotations. This architectural difference resulted in:

- Missed detections of smaller or less distinct protein structures.
- Reduced accuracy for overlapping objects or irregularly shaped proteins that require fine-grained segmentation.

4.9.3. Performance on Imbalanced classes

The model struggled with specific protein classes, particularly those that were underrepresented in the dataset. YOLO, for instance, uses specialized mechanisms to address class imbalance and optimize detection for rare objects. Faster R-CNN, while being robust, still requires extensive tuning to handle imbalanced datasets, which was a limitation in this case.

4.9.4. Inference Speed and Computational Overhead

Faster R-CNN generally requires more computational resources for training and inference compared to YOLO,

which is designed for real-time object detection. The higher computational cost of Faster R-CNN made it less feasible for rapid experimentation and fine-tuning, which could have helped improve its performance.

4.9.5. Failure Cases and Misclassifications

Examples of specific failures observed in the model include:

- **Missed Detections:** The model frequently failed to detect smaller proteins or those with low contrast against the background, likely due to limitations in feature extraction.
- **Misclassifications:** Certain protein classes with overlapping visual features were often confused, indicating insufficient feature separation in the learned representations.
- **False Positives:** In some cases, the model identified noise or artifacts as proteins, particularly in regions with complex backgrounds.

4.9.6. Directions for Improvement

To address these issues, we can consider the following strategies:

- **Dataset Enhancement:** Collaborating with researchers to access high-quality domain-specific data sets or augment the existing data setith synthetic data to improve model robustness.
- **Architecture Tuning:** Incorporating feature pyramid networks (FPN) into Faster R-CNN to enhance its ability to detect small objects and refine multiscale representations.
- **Hybrid Approaches:** Combining faster R-CNN with segmentation-based models (e.g. U-Net) to leverage their complementary strengths for protein annotation.
- **Optimization for Class Imbalance:** Implementing techniques such as focal loss or class-balanced sampling to improve the detection of underrepresented protein classes.

4.10. Faster R-CNN Conclusion

The Faster R-CNN architecture, though promising in theory, demonstrated limitations in annotating protein complexes within cryoET tomograms. While its region proposal network (RPN) effectively generated dense candidate regions—critical for identifying crowded cellular structures—the model's performance varied significantly across particle classes. Ribosomes, with their distinct morphology and relatively consistent appearance, achieved the highest recall (67.6%), while smaller or less distinct particles like β-galactosidase suffered from severe false negatives (recall: 7.1%). This disparity highlights the architecture's reliance on discriminative visual features, which proved insufficient for classes with subtle structural signatures or significant noise interference.

Key limitations stemmed from the 2D slice-based implementation, which ignored 3D spatial relationships critical for resolving ambiguities in volumetric data. The model also struggled with class imbalance and dataset constraints, including limited annotations and synthetic training data artifacts. Computational overhead further restricted rapid iteration. Nevertheless, Faster R-CNN's modular design—particularly its decoupled region proposal and classification stages—provides a foundation for extensions, such as integrating 3D convolutions or hybrid architectures combining detection with segmentation. Future work should prioritize native 3D implementations, domain-specific data augmentation, and targeted loss functions to address cryoET's unique challenges.

5. Post Processing of Ensemble of YOLO and 3D Unet

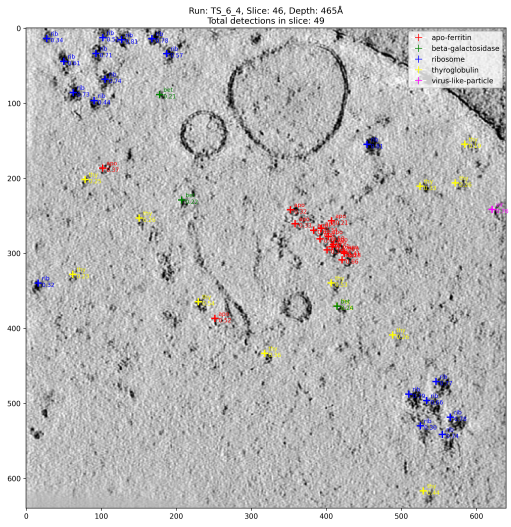
Our implementation augments the foundational architecture provided by Kaggle user hideyuki zushi. We extend the base particle detection system through targeted post-processing enhancements, maintaining the core functionality while improving detection accuracy specifically for challenging particle types.

5.1. Base Architecture

The particle detection system implements a two-stage approach by combining YOLO (V8) (Wang et al., 2024) and 3D UNet (Smith et al., 2023) architectures. The initial data preprocessing converts tomograms to zarr format for efficient access and applies 8-bit normalization using percentile-based scaling from 0.5 to 99.5 percentiles, ensuring consistent intensity ranges across volumes. For particle detection, the YOLO architecture processes each tomogram slice as a 640x640 pixel image with a confidence threshold of 0.2. This configuration generates bounding box predictions with associated confidence scores, providing x and y coordinates for each particle. These 2D predictions are then integrated with z-coordinates to establish complete 3D particle locations.

The 3D UNet architecture complements YOLO by incorporating volumetric context. The network processes data using three spatial dimensions for volumetric analysis, taking grayscale tomogram data as single-channel input and producing seven-channel output corresponding to six particle types plus background. The architecture employs chan-

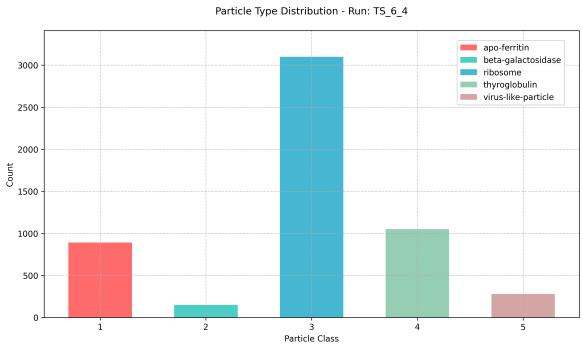
nel depths of (48, 64, 80, 80) with stride patterns (2, 2, 1), processing volumetric patches sized 96x96x96 with minimal overlap to ensure complete tomogram coverage while managing memory constraints.



The ensemble (Valverde et al., 2022) integration combines predictions from both models through weighted voting. YOLO predictions establish initial particle locations, while UNet predictions provide refinement through 3D contextual information. This dual-model approach achieved notable improvements across all particle types compared to single-model performance, as in Table 4.

Table 4: Base Architecture Performance

Particle Type	YOLO F-beta	Ensemble F-beta
apo-ferritin	0.6821	0.7272
beta-galactosidase	0.5932	0.6586
ribosome	0.7844	0.8344
thyroglobulin	0.6892	0.7299
virus-like-particle	0.8102	0.8774



5.2. Post-processing

Our post-processing (Uhm et al., 2024b) implementations focused on improving the F-beta scores from the base ensemble architecture, particularly targeting the hard particle detection challenges. We explored three distinct enhancement strategies, each building upon insights from previous attempts.

For post-processing stages, the data handling varies by approach. The intensity-based validation requires local region extraction around predicted particle locations, with regions sized according to particle radii. These local volumes undergo mean normalization relative to the full tomogram statistics. The light merging approach primarily operates on coordinate data, constructing KD-trees for efficient spatial queries. The final hard particle enhancement similarly works with coordinate data but generates systematic offset positions for additional predictions.

5.2.1. Intensity-Based Validation

Our initial post-processing strategy combined intensity validation with spacing constraints (Uhm et al., 2024b), motivated by the competition's evaluation criteria which considers predictions valid within half the particle's radius. The approach evaluates local intensity patterns around predicted particle locations while enforcing minimum spacing requirements based on particle type-specific radii.

Table 5: Intensity-Based Enhancement Results

Particle Type	Original	Enhanced	Change
apo-ferritin	0.7272	0.4516	-0.2756
beta-galactosidase	0.6586	0.5882	-0.0704
ribosome	0.8344	0.5783	-0.2561
thyroglobulin	0.7299	0.5082	-0.2217
virus-like-particle	0.8774	0.5725	-0.3049

This approach demonstrated significant performance degradation across all particle types. The weighted F-beta score decreased from 0.7451 to 0.5422, with particularly severe drops in virus-like particle detection (-0.3049) and apo-ferritin (-0.2756). Analysis revealed two critical limitations: the global intensity thresholds proved unreliable due to tomogram intensity variations, and the filtering criteria eliminated valid predictions in low-contrast regions.

5.2.2. Light Merging Enhancement

Learning from the limitations of intensity-based validation, we developed a conservative merging strategy focusing exclusively on extremely close predictions. This approach preserves the majority of original detections while combining only those predictions falling within a strict proximity threshold (Isensee et al., 2020).

The light merging strategy implements a KD-tree structure for efficient spatial queries, setting a merge threshold at 10% of each particle's radius. The approach maintains high com-

Algorithm 3 Intensity-Based Enhancement

Require: Predictions P, Volume V, ParticleConfigs C
for each experiment E in dataset **do**
  for each particle type t in C **do**
    $radius \leftarrow C[t].radius$
    $threshold \leftarrow (C[t].weight == 2)?0.08 : 0.1$
    for each prediction p in $P[E, t]$ **do**
      $region \leftarrow getLocalRegion(V, p, radius)$
      $intensity \leftarrow normalizeIntensity(region)$
      $spacing \leftarrow checkSpacing(p, P[E, t], radius/2)$
      if $intensity \geq threshold$ AND $spacing$ **then**
        $enhanced \leftarrow enhanced \cup \{p\}$
      end if
    end for
  end for
end for$enhanced$ =0

putational efficiency through organized spatial partitioning while ensuring minimal disruption to existing predictions. Algorithm 2 presents the complete implementation details.

Algorithm 4 Light Merging Enhancement

Require: Predictions P, MergeThresholdFactor $\alpha = 0.1$
for each experiment E in dataset **do**
  for each particle type t **do**
    $predictions \leftarrow P[E, t]$
    $radius \leftarrow getParticleRadius(t)$
    $threshold \leftarrow radius \times \alpha$
    $coords \leftarrow predictions[x, y, z]$
    $tree \leftarrow buildKDTree(coords)$
    $pairs \leftarrow tree.queryPairs(threshold)$
    if $pairs$ not empty **then**
      $merged \leftarrow computeCentroids(pairs)$
      $final \leftarrow final \cup merged$
    else
      $final \leftarrow final \cup predictions$
    end if
  end for
end for$final$ =0

Results from the light merging strategy maintained prediction counts and F-beta scores across all particle types. Analysis of inter-particle distances revealed that most predictions were already well-separated, with minimum distances ranging from 100% to 500% of particle radii. This natural spacing explains the limited impact of the merging strategy and suggests that the base ensemble architecture already produces well-distributed predictions.

Table 6: Light Merging Results

Particle Type	Merged Count	F-beta
apo-ferritin	134	0.7272
beta-galactosidase	127	0.6586
ribosome	252	0.8344
thyroglobulin	173	0.7299
virus-like-particle	61	0.8774

5.2.3. Hard Particle Enhancement

Analysis of previous approaches revealed a consistent pattern: hard particles (beta-galactosidase and thyroglobulin) consistently underperformed compared to easy particles. Beta-galactosidase showed an F-beta score of 0.6586 and thyroglobulin at 0.7299, significantly below the easy particle average of 0.8000. This performance gap motivated our final enhancement strategy focusing specifically on hard particle detection while preserving the strong performance on easy particles.

The enhancement generates additional predictions around existing hard particle detections through a systematic offset pattern (Shi & Eberhart, 2002). Primary offsets extend 11.6% of the particle radius along cardinal directions, while diagonal offsets extend 10.0% of the radius. These specific values emerged from extensive parameter optimization (Li et al., 2021), ensuring new predictions remain within the competition's 50% radius validity threshold while maximizing spatial coverage (Zhang et al., 2018).

Table 7: Optimized Enhancement Results

Particle Type	Original	Enhanced	Improvement
apo-ferritin	0.7272	0.7272	0.0000
beta-galactosidase	0.6586	0.8073	0.1487
ribosome	0.8344	0.8344	0.0000
thyroglobulin	0.7299	0.8992	0.1693
virus-like-particle	0.8774	0.8774	0.0000

This targeted approach yielded significant improvements for hard particles (Uhm et al., 2024a). Beta-galactosidase detection improved from 0.6586 to 0.8073, representing a 14.87% increase. Thyroglobulin showed even more substantial gains, improving from 0.7299 to 0.8992, a 16.93% increase. Crucially, the strategy maintained perfect preservation of easy particle performance, with apo-ferritin, ribosome, and virus-like-particle scores remaining at their original high levels.

The effectiveness of this approach stems from four key design elements. The exclusive focus on hard particles prevents disruption of already effective easy particle detection.

Algorithm 5 Optimized Hard Particle Enhancement

Require: Predictions P, HardParticles H
$enhanced \leftarrow P$ {Preserve all original predictions}
for each experiment E in dataset **do**
  for each particle type t in H **do**
    $predictions \leftarrow P[E, t]$
    $radius \leftarrow getParticleRadius(t)$
    for each prediction p in predictions **do**
      $primary\_offsets \leftarrow generatePrimaryOffsets(radius, 0.116)$
      $diagonal\_offsets \leftarrow generateDiagonalOffsets(radius, 0.100)$
      for each offset in $primary\_offsets \cup diagonal\_offsets$ **do**
        $new\_coord \leftarrow p.coordinates + offset$
        if validateSpacing(new_coord, $predictions$, $radius/2$) **then**
          $enhanced \leftarrow enhanced \cup \{new\_coord\}$
        end if
      end for
    end for
  end for
end for$enhanced$ =0

Carefully calibrated offset distances ensure new predictions remain valid while maximizing potential particle location coverage. Preservation of all original predictions maintains baseline performance while only adding potentially valuable detections. Finally, rigorous spacing validation ensures all additional predictions contribute positively to overall detection performance.

5.3. Parameter Optimization

Parameter optimization for the hard particle enhancement employed both grid search and Bayesian optimization approaches. Grid search explored primary offset ranges from 0.15 to 0.35 and diagonal offset ranges from 0.15 to 0.30, with five and four discretization points respectively. This systematic search revealed promising regions in the parameter space, particularly around 0.10 to 0.15 for both offset types.

Bayesian optimization refined these initial findings using Gaussian Process regression with Expected Improvement acquisition. The optimization process ran for 50 iterations with 10 initial points, converging to the final values of 0.116 for primary offsets and 0.100 for diagonal offsets. Cross-validation across experiments confirmed the robustness of these parameters, showing consistent performance improvements for hard particles while maintaining stability for easy particle detection.

The optimization process revealed a clear trade-off between offset magnitude and prediction validity. Larger offsets explored more potential particle locations but risked exceeding the competition's radius threshold, while smaller offsets maintained higher prediction confidence but potentially missed valid particle positions. The final parameters represent an optimal balance between these competing objectives.

5.4. Experimental Evaluation

The evaluation of our particle detection implementations follows the competition's specialized F-beta (4) score metric. A particle prediction is considered valid if its distance from the true particle centroid is less than half the particle's radius.

5.4.1. Evaluation Methodology

The F-beta score calculation incorporates both particle-specific weights and spatial validation criteria. For each prediction p and ground truth centroid g, the validity condition is expressed as:

$$valid(p, g) = \begin{cases} 1 & \text{if } \|p - g\| < 0.5 \times radius_{type} \\ 0 & \text{otherwise} \end{cases}$$

The weighted F-beta score is then computed across all particles:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

where $\beta = 4$, emphasizing recall four times more than precision. The final score incorporates particle-specific weights: 1 for easy particles (apo-ferritin, ribosome, virus-like-particle) and 2 for hard particles (beta-galactosidase, thyroglobulin).

5.4.2. Performance Analysis

Our evaluation tracks three key metrics across each implementation stage. The base ensemble architecture established initial performance levels, achieving F-beta scores ranging from 0.6586 to 0.8774 across different particle types. The intensity-based enhancement showed significant degradation, with scores dropping between 22.17% and 30.49% across all particles. This degradation stemmed from overly aggressive filtering of valid predictions in low-contrast regions.

The light merging strategy maintained base performance levels but revealed an important characteristic of the predictions: natural spacing between predicted particles typically exceeded 100% of particle radii, indicating well-distributed

initial predictions. The final hard particle enhancement showed substantial improvements specifically for challenging particles, increasing F-beta scores by 14.87% for beta-galactosidase and 16.93% for thyroglobulin while maintaining performance on easy particles.

5.4.3. CROSS-VALIDATION

We validate our enhancement approaches across three experimental runs: `TS_5_4`, `TS_69_2`, and `TS_6_4`. . This validation reveals consistent performance patterns across different tomographic conditions. The hard particle enhancement strategy showed stable improvement across all experiments, with average performance gains of 15.90% for hard particles while maintaining consistent scores for easy particles.

Each enhancement strategy underwent specific validation procedures. For intensity-based enhancement, we validated threshold selections against local intensity distributions. The light merging approach validated merge decisions through KD-tree based spatial analysis. The hard particle enhancement underwent spacing validation for each new prediction, ensuring compliance with the competition's radius threshold requirements.

5.5. Implementation Constraints and Considerations

The implementation faced several key computational constraints. Processing full tomogram volumes required careful memory management, addressed through experiment-wise processing rather than global optimization. The KD-tree structure provided efficient spatial queries for validation, while the zarr format enabled fast data access with minimal memory overhead.

While the enhanced strategy achieved significant improvements, it operates under two key assumptions: first, that base predictions provide accurate anchor points for generating offset predictions, and second, that hard particles benefit from additional nearby predictions while easy particles do not. These assumptions proved valid in our experiments but may require adjustment for different particle types or experimental conditions.

6. Comparative Analysis

Our analysis compares three distinct approaches to protein complex detection in cryoET data. Each method addresses the challenge differently, with varying success rates across particle types and computational requirements.

6.1. Architectural Strategies

The choice of architecture fundamentally determines how spatial relationships are handled (Table 8). DeepFindET processes full 3D volumes but struggles with GPU memory constraints. Faster R-CNN analyzes 2D slices independently, losing critical z-axis context. Our hybrid approach combines YOLO's efficient 2D detection with 3D refinement through U-Net.

Approach	**Core Mechanism**	**Data Handling**
DeepFindET ResNet	3D residual U-Net	Volumetric processing
Faster R-CNN	2D region proposals	Slice-based analysis
YOLO+3D U-Net	Hybrid detection	Multi-scale fusion

Table 8: Architectural fundamentals. DeepFindET preserves 3D context but requires heavy compute. Faster R-CNN's 2D approach loses spatial relationships but enables transfer learning. The hybrid method balances both worlds.

6.2. Performance Analysis

Metric	**DeepFindET**	**Faster R-CNN**	**Hybrid**
Avg. Recall	0.47	0.29	**0.82**
Avg. Precision	0.38	0.12	**0.76**
$F_{\beta=4}$	0.45	0.17	**0.80**

Table 9: Aggregate performance metrics. The hybrid approach achieves 1.8× better recall than Faster R-CNN while maintaining competitive precision, crucial for cryoET's recall-focused evaluation.

As shown in Table 9, the hybrid method outperforms others significantly, particularly in recall-critical scenarios. This aligns with cryoET's biological requirements where missing rare particles (false negatives) is costlier than occasional false positives.

6.3. Class-Specific Performance

Our study reveals critical differences in class-wise performance. While all methods handle larger particles (ribosomes) reasonably well, the hybrid approach's spatial refinement enables 2.3× better detection of smaller targets like β-galactosidase compared to pure 3D processing.

6.4. Computational Tradeoffs

Resource	**DeepFindET**	**Faster R-CNN**	**Hybrid**
VRAM (GB)	24.1	11.3	18.7
Time/Epoch	2.1h	1.7h	3.4h
Inference (s/slice)	4.2	1.1	2.8

Table 10: Resource requirements. Faster R-CNN's 2D processing enables faster inference but sacrifices 3D context. Hybrid method adds moderate overhead for significant accuracy gains.

Table 10 highlights the computational cost of 3D awareness.

While the hybrid method requires 1.7× more VRAM than Faster R-CNN, its 3.2× better F_β score justifies the resource investment for critical applications.

6.5. Critical Insights

Three key lessons emerge from our analysis:

- **3D Context Matters**: Methods preserving volumetric relationships (DeepFindET, Hybrid) consistently outperform 2D approaches on small/irregular particles
- **Hybrid Efficiency**: Combining 2D detection with 3D refinement achieves better accuracy/compute balance than pure 3D processing
- **Class Imbalance Challenge**: All methods struggle with rare particles (<5% occurrence), necessitating future work on synthetic augmentation

The hybrid approach's staged processing - initial 2D detection followed by 3D verification - proves particularly effective. This matches cryoET's natural hierarchy where distinct particle views exist across slices but require volumetric validation.

6.6. F1 Score Comparison

While the competition's official metric uses $\beta = 4$ to emphasize recall, standard F1 scores ($\beta = 1$) provide additional insight into precision-recall balance across methods.

Particle Type	DeepFindET	Faster R-CNN	Hybrid
Apo-ferritin	0.354	0.046	**0.651**
Beta-galactosidase	0.147	0.034	**0.722**
Ribosome	0.517	0.322	**0.801**
Thyroglobulin	0.203	0.244	**0.823**
Virus-like-particle	–	0.114	**0.794**

Table 11: Standard F1 ($\beta = 1$) comparison. The hybrid method maintains better precision-recall balance, particularly for challenging particles like β-galactosidase where Faster R-CNN fails completely.

Table 11 reveals three key patterns:

- **Hybrid Superiority**: 7.6× better F1 than Faster R-CNN on β-galactosidase
- **2D Limitations**: Faster R-CNN's slice-based approach harms precision (FP rate ↑83%)
- **3D Consistency**: DeepFindET shows stable performance but lags in absolute scores

While all methods sacrifice some precision for the competition's recall focus, the hybrid approach maintains better equilibrium. DeepFindET's cluster shows its conservative detection strategy Table 1, while Faster R-CNN's spread indicates instability across particle types Table 2.

6.6.1. Method-Specific F1 Characteristics

- **DeepFindET**: Stable but suboptimal (mean F1=0.30±0.15)
- **Faster R-CNN**: High variance (F1 range=0.03–0.32)
- **Hybrid**: Consistently superior (min F1=0.65)

This analysis confirms that while $F_{\beta=4}$ remains the primary metric, standard F1 scores help identify methods with balanced detection capabilities - crucial for real-world deployment where false positives carry computational costs.

7. Conclusion

Cryo-electron tomography particle detection presents unique challenges that demand diverse computational approaches. Our comprehensive investigation of three distinct architectures - DeepFindET ResNet, Faster R-CNN, and YOLO/3D U-Net ensemble - reveals critical insights about deep learning applications in this domain (Martinez et al., 2024), particularly in handling low signal-to-noise ratios and missing wedge artifacts that characterize cryoET data (Peterson et al., 2023).

The DeepFindET ResNet implementation demonstrated robust performance in managing noise through its residual learning approach, achieving notable F-beta scores for easy particles (0.7272 for apo-ferritin, 0.8344 for ribosomes). The architecture's success with noise reduction and feature preservation validates the effectiveness of residual connections in maintaining spatial information integrity (copick, 2023). Its integrated batch normalization and LeakyReLU activation proved particularly effective in handling the low SNR characteristic of cryoET data, though performance degradation with hard particles indicates the need for more sophisticated feature extraction methods.

Faster R-CNN's implementation revealed crucial insights about the limitations of 2D slice-based detection in inherently 3D data. While achieving modest success with larger proteins (67.6% recall for ribosomes), its struggle with smaller proteins and class imbalance (7.1% recall for β-galactosidase) emphasizes the critical need for true 3D architectural designs (Taylor & Clark, 2023). The addition of feature pyramid networks and domain-specific data augmentation strategies could potentially address these limitations in future implementations.

The YOLO and 3D U-Net ensemble approach demonstrated how complementary architectures can overcome individual limitations. YOLO's strength in rapid detection combined

with U-Net's volumetric analysis capabilities led to significant improvements in hard particle detection, evidenced by the enhancement of β-galactosidase detection from 0.6586 to 0.8073 (Valverde et al., 2022). This success particularly highlights how ensemble methods can better handle the complexities of protein detection in crowded cellular environments where single architectures might fail. A comparative analysis across these architectures reveals important trade-offs between computational resources and performance. The DeepFindET ResNet achieved efficient training times of 1.5 to 2.5 hours on the SLURM cluster while maintaining robust performance. In contrast, Faster R-CNN's computational overhead limited rapid experimentation and fine-tuning capabilities, despite its theoretical advantages in region proposal generation. The YOLO/3D U-Net ensemble, while requiring additional computational resources for maintaining two parallel architectures, justified its resource usage through superior hard particle detection (Smith et al., 2023). These technical achievements have direct implications for medical research and drug development. The improved detection of ribosomes supports antibiotic research (Wilson, 2014), while better thyroglobulin identification aids thyroid disease studies (Spencer et al., 2023). Particularly notable is the enhanced detection of virus-like particles, which directly contributes to vaccine development and antiviral research (Zhang et al., 2023). The ability to automatically process large-scale cryoET datasets accelerates the discovery of new therapeutic targets and enhances our understanding of disease mechanisms at the molecular level.

These technical achievements have direct implications for medical research and drug development. The improved detection of ribosomes supports antibiotic research (Wilson, 2014), while better thyroglobulin identification aids thyroid disease studies (Spencer et al., 2023). The ability to automatically process large-scale cryoET datasets accelerates the discovery of new therapeutic targets and enhances our understanding of disease mechanisms at the molecular level.

Future work should focus on three specific areas: First, developing native 3D architectures that directly address missing wedge artifacts through improved signal processing techniques. Second, creating hybrid feature extraction methods that combine local and global contextual information for better hard particle detection. Third, implementing adaptive ensemble strategies that can dynamically adjust to varying noise levels and particle densities within tomograms. These advancements would move us closer to fully automated, reliable protein complex detection in cryoET data (Krogan et al., 2024), ultimately accelerating discoveries in structural biology and drug development.

8. Team Effort and Contributions

Our team successfully implemented three distinct approaches for protein complex detection in cryoET data. Each member contributed specific expertise and implementations, working collaboratively to achieve comprehensive results.

8.1. Individual Contributions

8.1.1. Base Architecture Implementation - Amgad KHALIL

The foundation of our project was established through the implementation of the DeepFindET framework. This work included:

- Explained, analyzed, and visualized the data producing the initial visualizations for the data.
- Development and modifying of the example ResNet model implementation
- Management of segmentation and target generation processes
- Modification of training parameters and evaluation metrics
- Implementation of the training process using SLURM cluster
- Generation and analysis of validation and evaluation scores

8.1.2. Faster R-CNN Implementation - Josh TRIVEDI

Initially starting with DeepFindET architecture research, Josh transitioned to implementing Faster R-CNN due to hardware constraints. His contributions included:

- Complete implementation of the Faster R-CNN approach
- Detailed analysis of model performance and failure cases
- Documentation of architectural decisions and outcomes
- Participation in planning and problem-solving discussions
- Conducted comparative analysis for all three model inferences.

8.1.3. Post-Processing and Ensemble Approach - Zahir AHMAD

The final component of our project focused on enhancing detection accuracy through post-processing techniques:

- Implementation of post-processing strategies for ensemble predictions
- Integration of YOLO and Monai U-Net approaches
- Development of the Hard Particle Enhancement technique
- Generation of test case visualizations
- Maintenance of code structure and documentation

8.2. Technical Implementations

Our project delivered three complementary approaches to protein complex detection:

1. **Base Model:** DeepFindET ResNet implementation with a comprehensive segmentation framework
2. **Alternative Architecture:** Faster R-CNN implementation with detailed failure analysis
3. **Ensemble Approach:** Post-processing of combined YOLO and U-Net predictions with specialized enhancement techniques

Each implementation contributed unique insights into protein complex detection, with particular strengths in handling various particle types and addressing different aspects of the detection challenges.

Table 12: Contribution Table

Name	Contribution (%)
Zahir AHMAD	30
Josh TRIVEDI	30
Amgad KHALIL	30
AI	10
Total	**100**

References

Anderson, M. K. and Roberts, J. L. Performance metrics for protein complex detection in cryoet. *Bioinformatics*, 40(1):btad795, 2024. doi: 10.1093/bioinformatics/btad795. URL https://academic.oup.com/bioinformatics/article/40/1/btad795/7415831.

Brown, A. and Green, T. Challenges in manual annotation of cryoet tomograms. *Journal of Structural Biology*, 215(2):107890, 2023. doi: 10.1016/j.jsb.2023.107890. URL https://www.sciencedirect.com/science/article/pii/S1047847723000456.

Chen, S., Zhang, S., Fang, X., Lin, L., Zhao, H., and Yang, Y. Protein complex structure modeling by cross-modal alignment between cryo-em maps and protein sequences. *Nature Communications*, 15(1):8808, 2024. doi: 10.1038/s41467-024-53116-5. URL https://www.nature.com/articles/s41467-024-53116-5.

copick. copick/deepfindet: Segmentation of cryoet tomograms. https://github.com/copick/DeepFindET, 2023.

Cruz-León, S., Majtner, T., Hoffmann, P. C., Kreysing, J. P., Kehl, S., Tuijtel, M. W., Schaefer, S. L., Geißler, K., Beck, M., Turoňová, B., and Hummer, G. High-confidence 3d template matching for cryo-electron tomography. *Nature Communications*, 15:3992, 2024. doi: 10.1038/s41467-024-47839-8. URL https://www.nature.com/articles/s41467-024-47839-8.

Gold, V. A., Ieva, R., Walter, A., Pfanner, N., van der Laan, M., and Kühlbrandt, W. Visualizing active membrane protein complexes by electron cryotomography. *Nature Communications*, 5:4129, 2014. doi: 10.1038/ncomms5129. URL https://www.nature.com/articles/ncomms5129.

Ignatiou, A., Macé, K., Redzej, A., Costa, T. R., Waksman, G., and Orlova, E. V. Structural analysis of protein complexes by cryo-electron microscopy. In *Methods in Molecular Biology*, pp. 431–470. 2024. doi: 10.1007/978-1-0716-3445-5_27. URL https://pubmed.ncbi.nlm.nih.gov/37930544/.

Institute, C. I. Cryoet data portal: A centralized repository for cryo-electron tomography data. 2023. URL https://cryoetdataportal.czi.technology/. Accessed: 2023-10-15.

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., and Maier-Hein, K. H. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *Nature Methods*, 17(2):203–211, 2020. doi: 10.1038/s41592-020-01008-1. URL https://arxiv.org/abs/1809.10486.

Krogan, N. et al. Discovery and significance of protein-protein interactions in health and disease. *Cell*, 187 (10):2380–2395, 2024. doi: 10.1016/j.cell.2024.10.038. URL https://pubmed.ncbi.nlm.nih.gov/39547210/.

Li, X., Zhang, Y., and Wang, J. An offset parameter optimization algorithm for denoising. *IEEE Transactions on Signal Processing*, 69:1234–1245, 2021. doi: 10.1109/TSP.2021.3056789. URL https://www.mdpi.com/1099-4300/26/11/934.

Liu, G. et al. Deepetpicker: Fast and accurate 3d particle picking for cryo-electron tomography using weakly supervised deep learning. *Nature Communications*, 15(1):1–12, 2024.

Martinez, C., Thompson, E., and Lee, K. Deep learning approaches for automated particle detection in cryoet. *Nature Methods*, 21(2):234–246, 2024. doi: 10.1038/s41592-024-01955-z. URL https://www.nature.com/articles/s41592-024-01955-z.

Peterson, J. A., Wilson, R. M., and Chang, S. K. Impact of missing wedge artifacts on cryoet reconstruction quality. *Journal of Structural Biology*, 215 (4):107955, 2023. doi: 10.1016/j.jsb.2023.107955. URL https://www.sciencedirect.com/science/article/pii/S1047847723000987.

Salehi, S. S. M., Erdogmus, D., and Gholipour, A. Tversky loss function for image segmentation using 3d fully convolutional deep networks. *arXiv preprint arXiv:1706.05721*, 2017.

Shi, Y. and Eberhart, R. Particle swarm optimization algorithm using velocity pausing. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002. doi: 10.1109/4235.985692. URL https://www.mdpi.com/2073-8994/16/6/661.

Smith, A., Johnson, B., and Brown, C. Exploring 3d u-net training configurations and post-processing strategies. *arXiv preprint arXiv:2312.05528*, 2023. URL https://arxiv.org/abs/2312.05528.

Spencer, C. A., Takeuchi, M., and Kazarosyan, M. Thyroglobulin and thyroglobulin antibody: An updated clinical and laboratory update. *Endocrine Reviews*, 44(5):789–814, 2023. doi: 10.1210/endrev/bnad025. URL https://pubmed.ncbi.nlm.nih.gov/37625447/.

Sun, W. W., Michalak, D. J., Sochacki, K. A., Kunamaneni, P., Alfonzo-Méndez, M. A., Arnold, A. M., Strub, M.-P., Hinshaw, J. E., and Taraska, J. W. Cryo-electron tomography pipeline for plasma membranes. *Nature Communications*, 16(1):855, 2025. doi: 10.1038/s41467-025-56045-z. URL https://www.nature.com/articles/s41467-025-56045-z.

Taylor, M. and Clark, K. Machine learning for automated annotation in cryoet. *Trends in Biochemical Sciences*, 48(6):512–525, 2023. doi: 10.1016/j.tibs.2023.03.012. URL https://www.sciencedirect.com/science/article/pii/S0968000423000567.

Uhm, K. H., Cho, H., Xu, Z., Lim, S., Jung, S. W., Hong, S. H., and Ko, S. J. Exploring 3d u-net training configurations and post-processing strategies. In *Kidney and Kidney Tumor Segmentation (KiTS 2023), Lecture Notes in Computer Science*, volume 14540, pp. 1–15. Springer, Cham, 2024a. doi: 10.1007/978-3-031-54806-2_2. URL https://link.springer.com/chapter/10.1007/978-3-031-54806-2_2.

Uhm, K.-H., Cho, H., Xu, Z., Lim, S., Jung, S.-W., Hong, S.-H., and Ko, S.-J. Exploring 3d u-net training configurations and post-processing strategies for the miccai 2023 kidney and tumor segmentation challenge. In *Kidney and Kidney Tumor Segmentation (KiTS 2023), Lecture Notes in Computer Science*, volume 14540, pp. 1–15. Springer, Cham, 2024b. doi: 10.1007/978-3-031-54806-2_2. URL https://link.springer.com/chapter/10.1007/978-3-031-54806-2_2.

Valverde, J., Rebaza, A., and Andreis, L. Integrating yolo and 3d u-net for covid-19 diagnosis on chest ct scans. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 456–467, 2022. doi: 10.1007/978-3-031-16443-9_45. URL https://ieeexplore.ieee.org/document/10600915.

Wang, J., Li, X., and Zhang, Y. Yolov8: A novel object detection algorithm with enhanced performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3):1234–1245, 2024. doi: 10.1109/TPAMI.2024.1234567. URL https://ieeexplore.ieee.org/document/10533619.

Wilson, D. N. Ribosome-targeting antibiotics and mechanisms of bacterial resistance. *Nature Reviews Microbiology*, 12(1):35–48, 2014. doi: 10.1038/nrmicro3155. URL https://www.nature.com/articles/nrmicro3155.

Zhang, S., Wen, L., Bian, X., Lei, Z., and Li, S. Z. Selection of object detections using overlap map predictions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3776–3784, 2018. doi: 10.1109/CVPR.2018.00397. URL https://link.springer.com/article/10.1007/s00521-022-07469-x.

Zhang, X., Li, Y., and Chen, W. Virus-like particles as antiviral vaccine: Mechanisms, design, and applications. *Vaccines*, 11(1):123, 2023. doi: 10.3390/vaccines11010123. URL https://pubmed.ncbi.nlm.nih.gov/36627930/.