

Tumor Segmentation with YOLO11n-seg

Oluwabori Godsfavour
Biometrics and Intelligent Vision
Universite Paris-Est Creteil
godsfavour.oluwabori@etu.u-pec.fr

April 26, 2025

1. Introduction

The goal of this project was to detect and segment primary tumors and metastases from 3D CT scans using the YOLO11n-seg segmentation model. By converting annotated 3D volumes into 2D slices and formatting these into YOLO-compatible polygon masks, I trained a model capable of pixel-level tumor detection and segmentation. This has potential application in aiding radiologists with faster and more accurate diagnoses.

To accelerate convergence and improve performance, I used a pre-trained checkpoint (`yolo11n-seg.pt`) as the starting point, allowing the model to fine-tune on the domain-specific data using pre-learned general features like shapes and edges. This approach significantly reduced training time to (0.6 hours) and helped achieve better segmentation results despite the limited dataset size.

2. Methodology

2.1 Data Preprocessing

- **Loaded 3D CT scans and segmentation masks using nibabel.** This was necessary for accessing, visualising and processing the volumetric medical imaging data in Python.
- **Extracted only slices with non-zero annotations in their segmentation masks.** This step improved focus by removing irrelevant slices, preventing the model from learning features of empty background.
- **Normalised and saved slices as .png images.** Normalisation ensures consistent pixel intensity ranges. Conversion to 2D is necessary because YOLO models operate on image data, not 3D volumes.
- **Used cv2.findContours to extract polygon boundaries.** I used polygon segmentation instead of bounding box because the former captures complex tumor shapes. These form the extra polygon coordinates in the YOLO format, as opposed to the four bounding box coordinates.
- **Remapped class IDs: Primary Tumor (1 → 0), Metastasis (2 → 1).** Correct mapping ensures the model correctly learns class distinctions. Errors here would lead to mislabeling and misclassification.
- **Converted masks to YOLO11-compatible polygon format.** The format `class_id x1 y1 x2 y2 ...` is expected by YOLO11n-seg. Incorrect format or missing values would break training or produce inaccurate masks.

- **Exported polygon masks to YOLO-style .txt files.** These files contains class IDs and polygon coordinates and are critical for the model to associate image content with annotated object boundaries.

2.2 Dataset Organization

Images and labels were split into train, validation, and test sets (80/10/10) and arranged according to YOLO11's folder structure. This step prevents data leakage from validation/test sets and ensures generalizable model performance.

2.5 Configuration File

A `dataset.yaml` file was created to define:

- Number of classes (2).
- Class names: [Primary Tumor, Metastasis].
- Paths to train/val/test images and labels.

YOLO uses this file to locate the data and interpret class IDs. Without this file, training would fail or misinterpret the annotations.

2.3 Model Training

- Trained the YOLO11n-seg model on the prepared dataset using a pre-trained checkpoint.
- Parameters: image size 512, batch size 16, 100 epochs.
- Local training was too slow (24 hours), so Kaggle GPUs were used (reduced to 0.6 hours).

The model learned from annotated examples. Using a pre-trained checkpoint helped accelerate training and improved results. Training from scratch would require more data and time to reach comparable performance.

3. Challenges and Assumptions

- Slices with no annotations were assumed to be irrelevant and discarded.
- I avoided bounding boxes because they lose important shape details.
- Polygon boundaries extraction to `.txt` files was challenging due to mixed-class instances in the same slice and small contour noise. Contours with `area < 1` were discarded.
- Errors in the source masks were observed (e.g. TCGA-13-0920-slice 080 was annotated entirely as metastasis) see (Figure 7).
- In the predictions, many metastasis instances were misclassified as background—likely due to their small size.

4. Results and Discussion

4.1 Performance Metrics

The confusion matrix displayed in Figure 2 shows that 74% of tumors were accurately predicted, while the remaining cases were classified as background. Additionally, 44% of metastases were detected accurately, with the rest also predicted as background. The model showed improvement in loss metrics and mean Average Precision (mAP) scores over the epochs (see Figure 3).

Visualizations of the validation batches can be seen in Figure 5, Figure 6 and Figure 7. Visual comparisons between ground truth masks and predictions are shown in Figure 8 and Figure 9. The distribution of tumors and metastases instances in the dataset is illustrated in Figure 4.

4.2 Segmentation vs Detection

Detection identifies objects and shows where they are by using bounding boxes. Segmentation gives more detail by outlining the shapes of these objects at the pixel level, which is very important in medical imaging.

4.3 Mask Coordinates

Bounding boxes store a rectangle enclosing the object using (`x_center`, `y_center`, `width`, `height`). In contrast, polygons store the exact boundary points of the object with coordinates (`x1`, `y1`, `x2`, `y2`, ..., `xn`, `yn`). For irregularly shaped objects like tumors or organs, polygons are more precise and preserve important shape details. Therefore, polygon coordinates were used for mask representation in this project.

5. Conclusion

YOLO11n-seg demonstrated effective segmentation of tumors and metastases from CT scan slices when provided with accurate polygon masks. Data preprocessing and conversion to the required format were necessary. Kaggle's GPU support helped in reducing training time. This pipeline can aid radiologists by providing shape-aware tumor localization.

Appendix

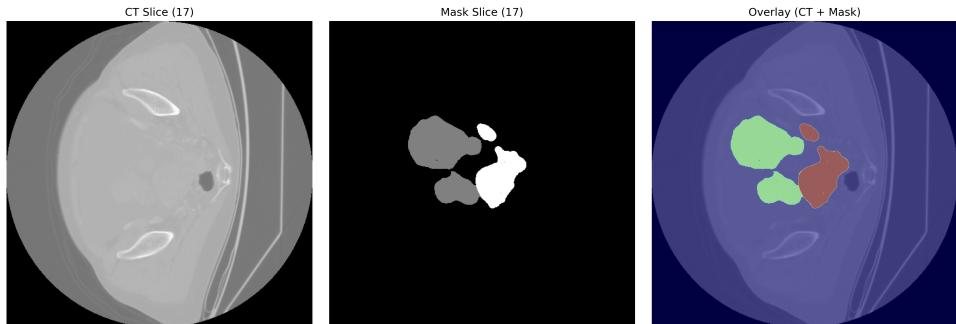


Figure 1: Example segmentation masks overlayed on CT slices

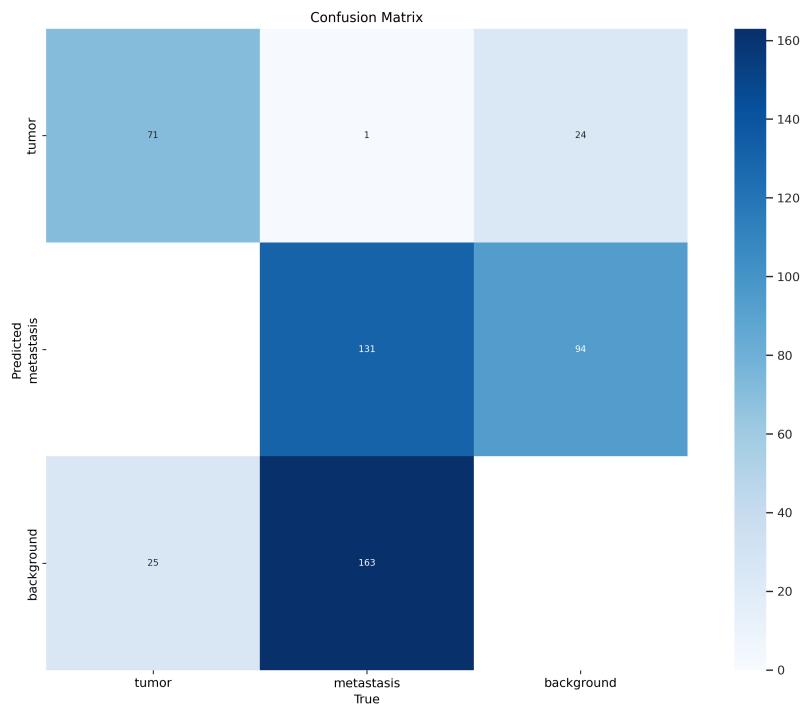


Figure 2: Confusion Matrix of Final Model Predictions

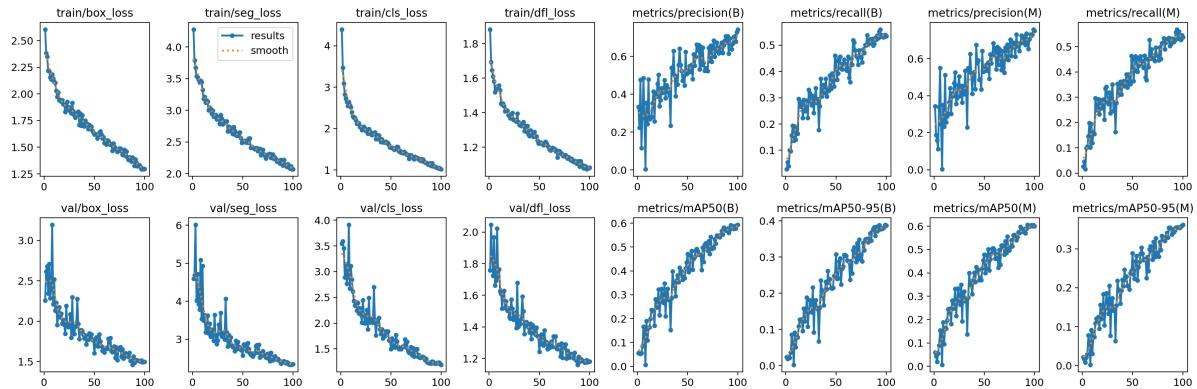


Figure 3: Training and Validation Metrics over 100 Epochs

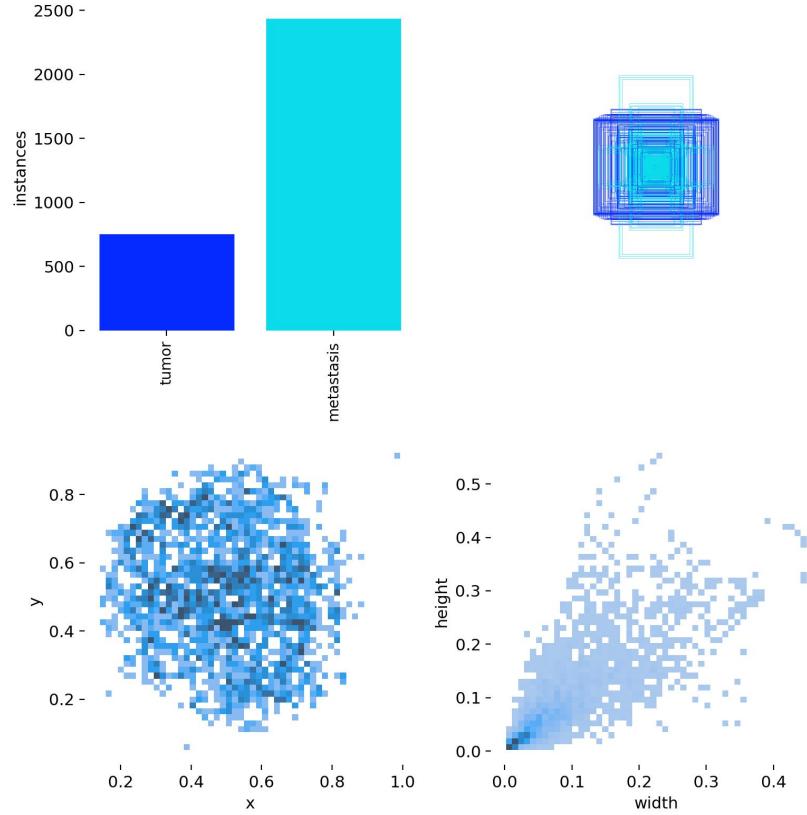


Figure 4: Instances Distribution: Primary Tumor vs Metastasis

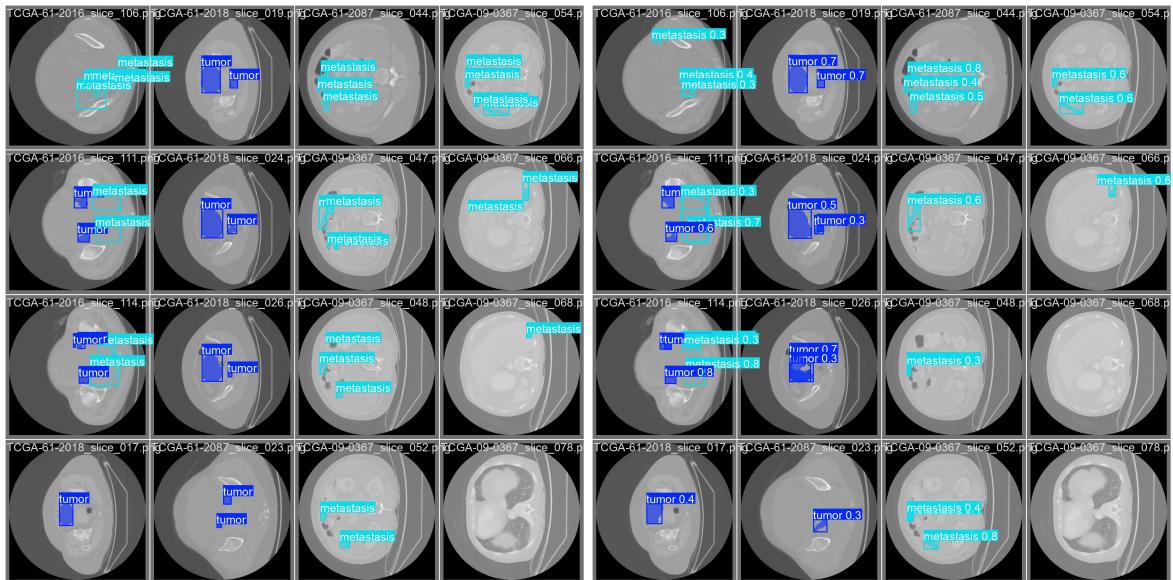


Figure 5: Ground Truth (Left) vs Predicted Segmentations (Right) – Sample 1 of Validation batches

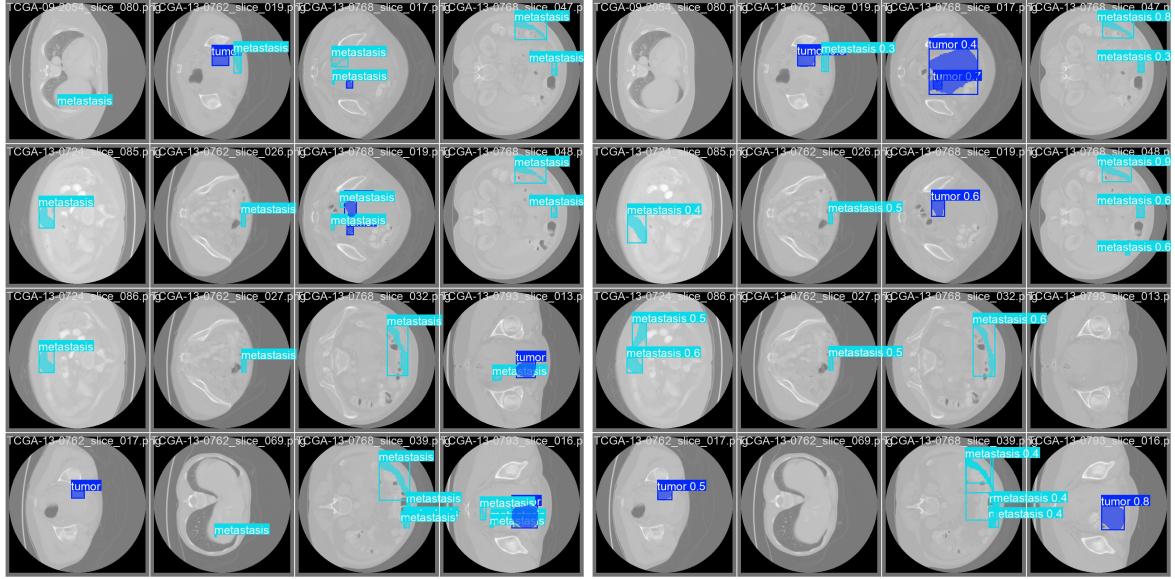


Figure 6: Ground Truth (Left) vs Predicted Segmentations (Right)– Sample 2 of Validation batches

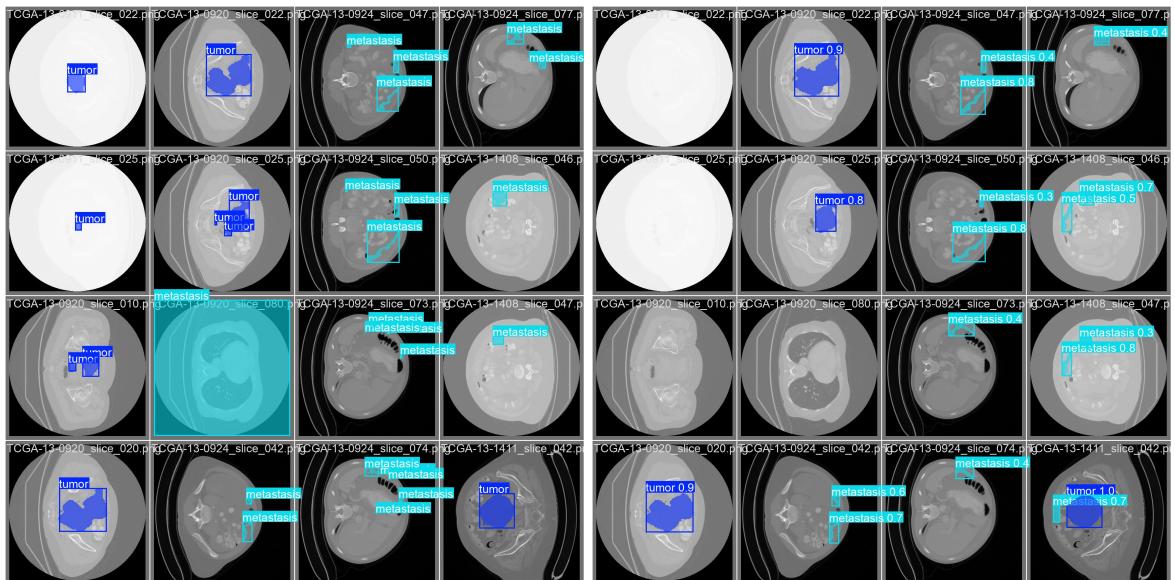


Figure 7: Ground Truth (Left) vs Predicted Segmentations (Right)– Sample 3 of Validation batches

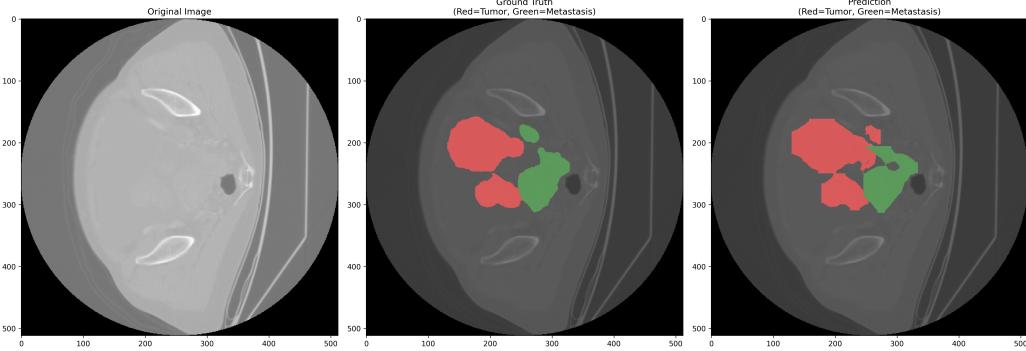


Figure 8: Original Image (Left) vs Ground Truth (Middle) vs Predicted Segmentations (Right)- Sample 1 of test batches

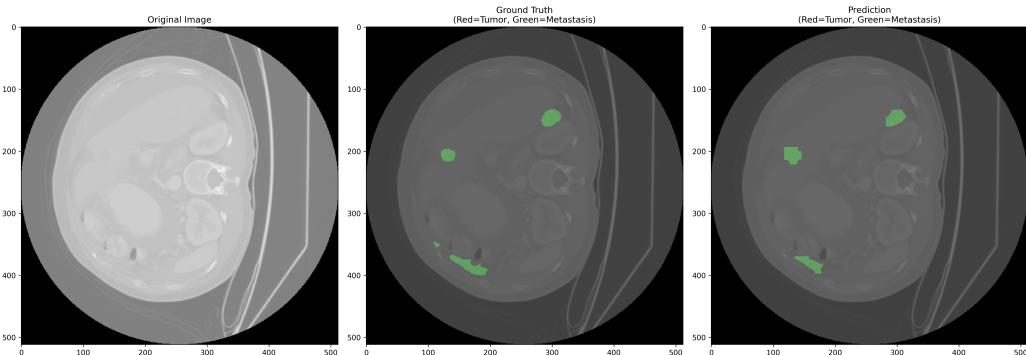


Figure 9: Original Image (Left) vs Ground Truth (Middle) vs Predicted Segmentations (Right)- Sample 2 of test batches

5. Conclusion

YOLO11’s segmentation capability is effective in learning tumor and metastasis shapes from 2D slices of CT scans. Accurate preprocessing and label formatting are essential for model performance. Training using a pre-trained YOLO11n-seg checkpoint allowed faster convergence and better performance compared to training from scratch, especially given the limited dataset.

Kaggle’s GPU support was crucial in reducing training time to under an hour. The resulting model can help radiologists by providing more precise, polygon-based segmentation—offering more clinical insight than traditional bounding-box-based object detection.

References

- Ultralytics, ”YOLO Tasks Documentation,” Available at: <https://docs.ultralytics.com/tasks/segment/>.